Consumer and Corporate Affairs Canada | Consommation et Corporations Canada

(11) (A) No. 1 188 811

(45) ISSUED 850611

(52) CLASS 354-121

(51) INT. CL. ^{3}G11C 15/00

(19) (CA) **CANADIAN PATENT** (12)

(54) Method and Apparatus for Information Storage and Retrieval

(72) Dissly, Donald D.;
Blanchard, Ronald J.,
U.S.A.

(73) Granted to Volt Delta Resources, Inc.
U.S.A.

(21) APPLICATION No. 217,334

(22) FILED 750103

(30) PRIORITY DATE U.S.A. (434,207) 740117

No. OF CLAIMS 80

Canada

DISTRIBUTED BY THE PATENT OFFICE, OTTAWA.
CCA-274 (11-82)

DEMANDES OU BREVETS VOLUMINEUX

LA PRÉSENTE PARTIE DE CETTE DEMANDE OU CE BREVET COMPREND PLUS D'UN TOME.

CECI EST LE TOME 1 DE 2

NOTE: Pour les tomes additionels, veuillez contacter le Bureau canadien des brevets

JUMBO APPLICATIONS/PATENTS

THIS SECTION OF THE APPLICATION/PATENT CONTAINS MORE THAN ONE VOLUME

THIS IS VOLUME 1 OF 2

NOTE: For additional volumes please contact the Canadian Patent Office

SPECIFICATION

This invention generally relates to the art of information storage and retrieval with special emphasis on the retrieval aspects of an overall information storage and retrieval system.

The problem and art of retrieving or selecting particular desired records from a set of stored records is an old one and must date back to early times when information bearing records began to be accumulated and stored in sets.

Perhaps one of the oldest and better known techniques is to organize the record set itself in some predetermined manner related to the expected retrieval process so that the record set can itself be manipulated to locate desired records. One simple example of such a system would be the segregation of documents in a typical household filing system so that records relating to current bills are in one file folder, those relating to this year's tax records are in another file folder, etc. In such a system, one may access or retrieve particular documents or records by first sorting through all of the possible catagories of segregated documents and selecting those which would probably be most closely related to the particular document desired and then physically manipulating each record in those particular segments of the overall collection of records until the particular desired document is discovered.

In this technique, retieval accuracy is something less than 100% (unless every record of the entire set is

always searched) and retrieval precision is obviously a matter of chance.

Another variation in prior known storage and retrieval systems involving hierarchical organizations of the information bearing records would be the alphabetical organization of topics such as are found in the usual encyclopedia. Here, one must first identify a "key word" relating to the topic of interest and then access the key word organized data file with his knowledge of the alphabet to locate that particular section of the file relating to that particular key word. Depending upon the complexity and extent of information that is to be retrieved in this manner, one or more such manipulations of the information bearing records may suffice.

However, as the complexity of the information stored on each individual record increases and as the number of records increase and as the complexity of the desired accessibility of such records increases, retrieval methods requiring direct manipulations of the information bearing records become cumbersome, time-consuming and counter-productive. For instance, one may often want to retrieve particular records from the data file having some common characteristics that were never before considered to be particularly relevant or important and which were not specifically taken into account when the records were first organized and stored. Thus, there is a need for a more generalized method of retrieval that does not involve the

the necessity of manipulating the entire base data file each time one wants to search that data file for particular records having desired predetermined characteristics that might never have before been considered important.

To help meet such needs in the past, many different approaches have been attempted and some have worked with various degrees of success for particular applications. For instance, we are all familiar with the device commonly known as an index whereby the subject matter or information contained in a set of records is carefully classified according to a predetermined set of topics. The topics included in the index may of course be quite detailed and, in fact, there is a whole science of taxonomy which can be called upon to help in organizing the hierarchical structure of such an indexing arrangement. However, in practical effect, the index is just a more sophisticated form of the earlier discussed more primative system whereby the records themselves are physically segregated into sections related to particular topics. Of course, with an index system, the set of records is itself separate and apart from the index and is organized in some predetermined manner such as by sequential numbers, etc. To use such a system of information storage and retrieval, one must take his special and perhaps unique desire for information and attempt to fit it within the predetermined taxonomic organization of the preexisting index and attempt to locate those index topics most closely related thereto. A cross-reference may then be had to the particular page numbers or sequence numbers, etc., of the source documents themselves for access. Of course, since the

source documents themselves are not actually included within the index, it is possible to have reference at more than one place in the index to the same source document or documents. However, unless some anticipated and thus already indexed retrieval inquiry is involved, such index systems have obvious limitations.

Besides the simple indexes which are in daily use by the general public, there are, of course, much more sophisticated versions of such indexes used in computerized search and retrieval systems. However, all such known indexing systems are an inherent compromise between the optimum and the practical since the taxonomic organization of the index is necessarily fixed with respect to a whole population of information bearing records and thus not uniquely tailored to any one of them nor is the taxonomic organization of such an index uniquely related to the unknown future requirements of those who will use the index. At best, it is a compromise hopefully representing a usable practical interface between the user and the actual base data file of information bearing records.

Another known attempt to interface a large set of information bearing records with the ultimate user involves the so-called abstracting and/or key word systems wherein abstracts of each information bearing record and/or selected key words from the text of such information bearing records are organized in a smaller abstract or key word file which can be more quickly searched than the entire voluminous file of information bearing records. Often, the actual file of information bearing records are not in machine readable

form but are located in an archive someplace. Often, the abstract file or file of key words, etc., together with appropriate pointers to the actual location of corresponding information bearing records in the archive is in machine readable form. Thus, the abstract or key word file may be machine accessed so that the user has to merely supply special key words, etc., whereupon a computer search of the smaller abstract or key word file is initiated in the hope of locating particular corresponding records in the archive having those key words therein.

As can be appreciated, this technique actually involves physical manipulation of the abstract or key word file and thus if this file reaches significant proportions, the search itself may take a significant time and become cumbersome, etc., subject to the same infirmities as the most primative system wherein all of the information bearing records are themselves manipulated in order to search for and retrieve particular desired ones of those records. Furthermore, these systems are subject to additional infirmities in that the user of the system may not always associate the same key words with a particular information bearing record as did the particular person who wrote the abstract or abstracted the key words from the original text of the document in question. As will be appreciated, the abstract or key words associated with a particular document would no doubt be selected to represent what was then thought to be the significant aspects of that particular document.

However, as is often the case, some subsequent user may be searching for that same particular document for a quite different reason which would appear completely secondary and perhaps even unimportant in the context of earlier thought as to what was significant about that particular document.

Many manual, semi-automatic and even automatic systems have been devised to aid in the art of information storage and retrieval in the past. For instance, U.S. Patent No. 3,354,467 issued in 1967 to Beekley shows a machine for automatically comparing superimposed binary coded tapes wherein each tape represents some particular predetermined characteristic potentially associated with particular documents or information bearing records in an archive file. A selected set of such tapes representing a desired set of such predetermined characteristics is selected and they are simultaneously fed through a photo-electric scanning station such that only those documents having all of the desired predetermined characteristics are identified as to location within the archive file. A somewhat similar semi-automatic or even manual indexing system was involved in the system earlier offered by McBee Systems wherein an abstract card or the like was coded with notches around the periphery of the card and then selected by "pinning" those cards having holes instead of notches at some particular location around the periphery of the card.

However, it will be noted that none of these prior systems have utilized the basic language structure of the

information contained on the set of information bearing records to construct a retrieval file which is uniquely custom fitted to each and every one of the documents or records in the base data file and which can therefore be utilized by a user in a manner which is uniquely and custom fitted to his peculiar information retrieval requirements without regard to whether or not those information retrieval requirements necessarily fit within the taxonomic organization of some preexisting index system. It is an object of this invention to provide such a unique and custom fitted retrieval file capability.

That is, the retrieval file of this invention is structured so as to optimally interface between a base data file and a user of that file. It takes into account the individualized language structure for the information content of each record in the base data file. The basic coding or generation of the retrieval file for this invention does not require the classification of the information content as being of this particular type or as of that particular type, although such traditional taxonomic classification may be conveniently incorporated within the retrieval file of this invention if desired.

Rather, in the exemplary embodiments to be described in more detail below, the basic retrieval file is coded so as to take into account the language structure of the information content of the records. For instance, the alphabetic value of informational characters in each record and the relative sequential location of such character values in associated groups of characters such as words in

those records are utilized in one preferred exemplary embodiment. Thus, although the resulting retrieval file is irreversible (except in the most simple cases) it is still uniquely representative of the language structure used in the entire informational content of the corresponding information bearing record. If one wants to think of the system of this invention in terms of "key words", then every word in every record would constitute a "key word" in that the language structure of all such words would be coded in the retrieval file. Thus, the user is not restricted to some other persons prior choice of "key words".

Of course, if desired, one may combine the retrieval file and/or other teachings of this invention with earlier types of systems to provide a modified and greatly improved version of such earlier systems. For instance, a very large abstract or key word file itself may be utilized for generating a retrieval file according to this invention rather than using the full text of the information bearing records. In such a case, the retrieval file will reflect, of course, only the full information content of the abstract file and/or key word file from which it was coded. However, it will now be considerably more simple to search the retrieval file of this invention than to perform a sequential search of a more lengthy abstract or key word file, etc., thus making the marriage of the two systems a profitable one for certain applications.

In general, the retrieval file of this invention comprises a plurality of arrays of binary coded elements. Each such array is organized to include a binary coded element respectively corresponding to the address or location of each record in the base data file. In addition, each array in the retrieval file is assigned to correspond to a predetermined identifiable characteristic of language structure potentially present in or associated with such records. In one of the exemplary embodiments, those predetermined identifiable characteristics are themselves specially chosen to represent the alphabetic value and relative sequential location of informational characters in the text of the information bearing records in the base data file for associated groups of characters such as words, etc.

Each binary coded element in any given array is assigned a predetermined binary value to represent the presence or absence of the predetermined identifiable characteristic represented by the given array in the particular record represented by each element. In this manner, the arrays of binary coded elements comprising the retrieval file represent an irreversible data compression of whatever information from the records that has been used to generate the retrieval file. In the preferred embodiments, the full text of each record is utilized for such coding purposes so that the retrieval file itself represents an irreversible data compression of the entire full text of each record in the base data file. On the other hand, if only an abstract or key word set from the information bearing

records is utilized for coding a retrieval file according to this invention, then the retrieval file will likewise represent an irreversible data compression of this more limited amount of information from such records.

Once the retrieval file has itself been generated directly from the information bearing records or from extracts thereof, etc., the retrieval file may be utilized by identifying those predetermined identifiable characteristics (i.e. the particular arrays of the retrieval file) associated with desired search or retrieval inquiry data (i.e. any particular word or groups of words, etc., thought to be in the text of the sought after record or its extract, etc.). Those particular arrays are then selected and the binary values of respectively corresponding elements in the selected arrays are compared to identify which records (actually the location or addresses of such records) in the base data file have all the desired predetermined identifiable characteristics.

To help in understanding the basic functioning of the invention, it is helpful to reexamine some of the basic characteristics of language structures. Using the English language as an example, it is readily apparent that almost all written information involves considerable redundant usage of a very limited number of alphabetic character values. Of course, besides character values per se, there are other important features of our written language such as the relative sequence of character values, upper and lower case differentiation of the alphabetic character values, punctuation, bold face type, italicized type, the size of printed

type, etc. There are also word length differences (i.e. groups of characters associated together in different numbers).

On the typical page of a printed book there may be something on the order of 3,000 characters so that if each of the 26 alphabetic characters of our alphabet are equally used on such a page, there would be about 115 redundancies with respect to each alphabetic character if one were to look only at the alphabetic value of characters. Thus, it is clear that the real information conveying content in any document or record utilizing such a written language is critically dependent upon other language structures such as word length, character sequences, type case, hyphenation, etc.

Thus, this invention is directed to an information and storage and retrieval system wherein the retrieval file is custom fitted or keyed to the language structure of the information contained in each record of the data base file. At the same time, the retrieval file of this invention inherently masks language structure redundancies and does not necessarily reference all the information contained in such a document. In short, the exemplary embodiment of the retrieval file is organized so as to conveniently represent at least some of the alphabetic values of informational characters and their relative sequential location in associated groups of characters such as words contained in the records of the base data file. There are many possible ways to take the alphabetic values and sequential location of such values into account. However, in the exemplary

embodiment of this invention, the redundance of character value occurrences within words of various lengths has been utilized to greatly compress the storage of such character values and sequential location data in the retrieval file.

That is, for each letter of the alphabet, there is a specific position of occurrence for each possible character value for each possible word length. Thus, the letter "A" can be a single letter of a single letter word; it can be the first letter of a two-character word or the second letter of a two-character word; it can be the first letter of a three-character word, etc. Thus, a binary coded matrix of character value versus character position within words having particular numbers of characters can be formed and this process can be repeated for any word length. However, it has been found that it is not necessary to include such coding in the retrieval file for all word lengths.

In fact, depending upon the particular application involved, it may often be advantageous to simply make the matrix representative of character value versus character position within a word regardless of the word length. In effect, this would amount to the superposition of and hence further compression of data from the first discussed organization of the matrix. From a practical standpoint, even if the first mentioned more detailed matrix organization is used, it is usually only desirable to include data in the matrix with respect to character values up to 6 or 7 letter words. In other wrods, it is only necessary in the usual

cases to keep track of character values and/or character value sequences for the first few characters of each word to produce an acceptable and usable information retrieval capability based on the language structure of information in the base data file of records.

As an example of the information compression involved in such a system, it should be recognized that an entire set of such predetermined language structure characteristics up to the maximum word length of seven characters would involve only 728 binary bits (28 bits per character for relative position information times 26 possible character values) so that for information retrieval purposes, the entire information content of an information bearing record would, in this simple example, be compressed to 728 binary bits of information. Of course, redundancies in character values and sequential locations would be irreversibly lost in the coding process but, as will be seen in the more detailed description given below, this loss of redundancy will not seriously affect most usages of the retrieval file and whatever loss of precision that is caused by such an organization of the retrieval file can be compensated for by other techniques.

Another possible example of coding representations of the alphabetic value of informational characters and their relative sequential locations within words would be to note the occurrence of alphabetic values as even or odd sequential positions within even or odd length words. For instance, the occurrence of the letter "A" at an even

position within an odd word length; as an even position within an even word length; as an odd position within an odd word length; and/or as an odd position within an even word length, etc.

As will become more apparent, there are trade offs to be made in deciding how to structure the retrieval file of this invention. For instance, if more detailed and complete language structure information is maintained with respect to character value and/or sequential location within words, then more precision can be obtained in accessing the base data file. On the other hand, more arrays are needed in the retrieval file to maintain such detailed information.

As an example of some of the types of possible limitations of the exemplary embodiment of this invention with respect to the precision of retrieval, consider the following example. Assume that one of the documents in the base data file includes the word "BROWN" and the word "BLACK". Since these words are both five-character words in length, the initial B character value for each word represents a redundancy in information and this would be represented by a single binary bit in the retrieval file showing only that there was at least one word in this document having five characters total length and a first character of value "B". The remaining characters of each word would of course also be binary coded to represent their value and position sequence within these two five-character words. Thus, when the retrieval file is used for retrieval purposes, it would be absolutely 100 percent

accurate in that these particular documents would always be indicated as including the word "BROWN" or the word "BLACK" respectively. However, the precision of retireval is not necessarily 100 percent since there are at least the following three search words which would also result in the selection of that document: "BLOWN", "BRACK", and "BRAWN". That this is so can be seen since all are five letter words and include an initial letter character value of B. Furthermore, the word "BLOWN" is a five-character word having a second character value of L (this characteristic would have been coded because the word "BLACK" was coded from the original document); a third character value of "O" (this value would have also been coded because the word "BROWN" was found in the original document); a fourth character value of "W" (also resulting from the word "BROWN") and a fifth character value of "N" (also resulting from the word "BROWN".

Thus, so far as the user of the system is concerned, one would obtain responses from inquiries related to any of the five words BROWN, BLACK, BLOWN, BRACK or BRAWN although only the words BROWN and BLACK are actually found in the original documents. Accordingly, although the retrieval would be absolutely accurate (that is the required document would always be accessed whenever the words "BROWN" or "BLACK" were used for inquiry) it would not be 100 percent precise in that either or both documents might also be erroneously withdrawn in response to three other spurious inquiry words.

The degree of precision is, of course, a function of the particular algorithm used to chose the language structure characteristics such as alphabetic value and relative sequential location representations in the retrieval file. For instance, the further data compression achieved by taking into account only the odd-even word lengths and odd-even character positions and values therein would further compound such a "mis-hit" problem. On the other hand, other techniques such as the inclusion of upper and lower case as uniquely identifiable character values would increase the precision and restrict the number of improper possible combinations.

In short, although the retrieval technique of this invention is 100 percent accurate, its precision may possibly be less than that unless care is taken in choosing the language structure characteristics used in the retrieval file because some specific character sequence data contained in the original text is lost (in one exemplary embodiment) in the conversion to the retrieval file. This is, of course, necessary in order to reduce the amount of information in the retrieval file from that actually on the original documents. If all of the character value and sequence information were to be retained in the retrieval file, then the file would be reversible and would, in actuality, contain all of the information in the original document. However, since the retrieval file is itself used only to look up and find the location of the actual document in question, it is not desirable to

include the entire detailed informational content of the original documents in the retrieval file as should be apparent. Thus, in many cases, it will be necessary to tolerate some small controllable amount of precision less than 100 percent. In fact, in some cases, a lack of precision is desirable. For instance, when the user is not too sure exactly what is to be retrieved (a telephone directory assistance operator without full name or other information), a lack of precision retrieval may actually assist in quickly accessing the actual desired record.

In the exemplary embodiment to be described, each word of data or block of text in a record of the base data file is handled in a uniform manner. All hypens are considered as spaces, all apostrophes are considered as spaces, all other punctuation is ignored. Any capitalized word will be considered containing all capitalized letters and all words starting with lower case letters will be considered lower case throughout. Words containing both numerals and letters will have their word length determined by the total number of numerals and alphabetic characters. However, the position of each digit and alphabetic character will be assigned at their nominal positions within such an overall word.

The information storage and retrieval technique of this invention may be conveniently practiced manually, semiautomatically, automatically with special purpose equipment, automatically with specially programmed and

conditioned general purpose equipment, etc., as will be described in more detail below.

A more complete understanding and appreciation of this invention may be obtained by reading the following detailed description in conjunction with the accompanying drawings, of which:

FIGURE 1 is a schematic depiction of an exemplary embodiment of an information storage and retrieval system incorporating this invention;

FIGURE 2 is a more detailed schematic depiction of the binary coded retrieval file for the exemplary embodiment disclosed in FIGURE 1;

FIGURE 3 is a binary coded matrix showing explicitly the binary coding required for an exemplary embodiment of this invention with respect to a particular text reproduced in the detailed description given below;

FIGURE 4 is a composite schematic showing of a few of the arrays in an exemplary system of binary coded arrays corresponding to the binary coding shown in matrix form at FIGURE 3;

FIGURE 5 is a schematic diagram of an exemplary embodiment of apparatus for practicing an embodiment of this invention;

FIGURE 6 is a schematic diagram of another apparatus for practicing an exemplary embodiment of this invention;

FIGURE 7 is a block diagram of the conversion, file construction and retrieval processing for data in an exemplary embodiment of this invention;

FIGURE 8 is a more detailed block diagram of the conversion block shown in FIGURE 7;

FIGURE 9 is a more detailed block diagram of Program No. 1 shown in FIGURE 8;

FIGURES 10-17 are block diagrams of subroutines entered from Program No. 1 shown in FIGURES 8-9; Figure 10 being on the same sheet as Fig. 10, and Fig. 12 being on the same sheet as Fig. 13.

FIGURE 18 is a more detailed block diagram of Program No. 2 shown in FIGURE 8;

FIGURES 19-24 are block diagrams of subroutines entered from Program No. 2 shown in FIGURES 8 and 18;

FIGURE 25 is a more detailed block diagram of the file construction block shown in FIGURE 7;

FIGURE 26 is a schematic representation of the intermediate retrieval file format used in the retrieval file construction;

FIGURES 27-28 show the magnetic disk files organization for the exemplary embodiment of FIGURES 7 et. seq.;

FIGURES 29-32 together constitute a block diagram of Program No. 5 shown in FIGURE 25;

FIGURES 33-43 are block diagrams of subroutines entered from Program No. 5 shown in FIGURES 25 and 29-32;

FIGURE 44 is a more detailed block diagram of Program No. 3 shown in FIGURE 25;

FIGURES 45-52 are block diagrams of subroutines entered from Program No. 3 shown in FIGURE 44;

FIGURE 53 is a more detailed block diagram of Program No. 4 shown in FIGURE 25;

FIGURES 54-62 are block diagrams of subroutines entered from Program No. 4 shown in FIGURE 53;

FIGURE 63 is a more detailed block diagram of the retrieval processing block shown in FIGURE 7;

FIGURES 64-68 together constitute a block diagram of Program No. 6 shown in FIGURE 63;

FIGURES 69-84 are block diagrams of subroutines entered from Program No. 6 shown in FIGURES 64-68;

FIGURE 85 is a schematic/block diagram of a semi-automated exemplary embodiment of this invention;

FIGURES 86-90 are enlarged photographs of PIC arrays for $A_{1,1}$; $A_{1,2}$; $A_{2,2}$; $A_{1,3}$ and $A_{2,3}$ respectively for the exemplary embodiment of FIGURE 85;

FIGURE 91 is an enlarged photograph of a composite array formed by a Boolean AND operation on the arrays of FIGURES 87 and 88; and

FIGURE 92 is a block diagram generally showing the interrelationship of computer programs for use in the semi-automated FIGURE 85 embodiment to automatically construct the retrieval file arrays.

The following detailed discussion will be generally organized in three sections. The first section deals with generalized concepts and features of an exemplary embodiment of the invention with special emphasis on the binary coded retrieval file, its organization and content, etc. The second section will deal with a detailed description of a specific exemplary embodiment of the invention that has actually been tested on existing general purpose digital computing equipment for accessing telephone directory records in a machine readable base data file. Finally, the third section of this detailed description will relate to an exemplary embodiment of special purpose apparatus for practicing another exemplary embodiment of the invention.

GENERALIZED DISCUSSION OF AN EXEMPLARY EMBODIMENT

Referring to FIGURE 1, a base data file 100 of accessible records stored at known addresses is shown. Here, the addresses are indicated as 4 unique digits associated with each record and the first 4 documents are shown in FIGURE 1 together with a corresponding indication of their respectively corresponding 4 digit address numbers. This base data file may comprise an already existing archive of records; it may comprise a machine readable and machine accessible file of records such as might be stored on magnetic tape, magnetic disc, magnetic core, etc; it may comprise a collection of photographic images such as on microfilm or microfiche stored in conventional equipment such that any given document may be rapidly and automatically retrieved for display upon supplying the appropriate address

information, etc. In short, the base data file 100 may comprise any accessible file or records wherein each record may be referenced by some unique address, location, or other equivalent pointer.

This base data file 100 is then utilized directly as indicated at 102 to construct binary coded arrays comprising a retrieval file 104 of such binary coded arrays. Of course, as previously mentioned, modifications of the exemplary embodiment may also be made wherein only extracts such as abstracts or selected key words, etc., from each record are utilized at 102 to construct the retrieval file 104. Exemplary processes and apparatus for constructing the binary coded arrays as indicated at 102 will be further detailed below. Of course the arrays could also be manually constructed as will be apparent. The result of such a construction or generation process will be a plurality of arrays of binary coded elements some of which are schematically illustrated in FIGURE 1 within block 104. An enlarged and more detailed version of this portion of FIGURE 1 is shown in FIGURE 2.

It should be recognized that each array in the retrieval file corresponds to some predetermined identifiable characteristic of language structure (hereafter referenced merely as a "PIC"). Furthemore, there is an element or area of each array which corresponds to each and every one of the correspondingly associated record addresses in the base data file.

For instance, as may be seen more clearly in FIGURE 2 where element address numerals have been added for clarification, there is an element in each of the arrays which corresponds to the address of record numbers 0000; 0001; 0002; 0003; etc. The cross-hatched elements are the ones that have been shown, for purposes of illustration, as having received a binary valued code representing the presence of a corresponding PIC (corresponding to the particular array)in the respectively corresponding record. For instance, the record stored at location or address 0201 is indicated by the cross-hatching as including PIC #1 while the record located at address 0202 is indicated as having an opposite binary value or code indicating the absence of PIC #1 from that particular record.

Of course, the particular set of PIC's are chosen as a function of the language structure of the information contained in the base data file. In the exemplary embodiment, for retrieving from a base data file comprising English language written records such as newspaper clippings, photo captions from newspaper clippings, etc., the set of PIC's are chosen to represent the alphabetic value of informational characters in the records and to represent the relative sequential location of such character values in associated groups of characters such as words contained within the records. In particular, the PIC's are chosen to correspond to the alphatetic value of the first, second, et. seq. character positions in words having one, two et. seq. total characters therein.

This exemplary technique for choosing the set of PIC's may be illustrated easily by the following matrix representation wherein each PIC represents an entry in the matrix in the form of $\phi_{m,n}$ where ϕ equals the alphabetic value; m equals the character position number within a word and n equals the word length. In this exemplary embodiment, ϕ takes on the values a through z; 0 through 9; and A through Z:

Predetermined Identifiable Characteristic = $\phi_{m,n}$

ϕ = Alphabetic Value
m = Character Position No.
n = Word Length

$\phi_{1,1}\phi_{1,2}\phi_{1,3}\phi_{1,4}\phi_{1,5}\phi_{1,6}\phi_{1,7}\phi_{1,8}\phi_{1,9}\phi_{1,10}\phi_{1,11}\phi_{1,12}\phi_{1,13}\phi_{1,14}\phi_{1,15},\phi_{1,16},\phi_{1,17},\phi_{1,18}\phi_{1,19}\phi_{1,20}$ ...

$\phi_{2,2}\phi_{2,3}\phi_{2,4}\phi_{2,5}\phi_{2,6}\phi_{2,7}\phi_{2,8*}$ - - - - - - - - - - - - -

$\phi_{3,3}\phi_{3,4}\phi_{3,5}\phi_{3,6}\phi_{3,7}\phi_{3,8*}$ - - - - - - - - - - - - -

$\phi_{4,4}\phi_{4,5}\phi_{4,6}\phi_{4,7}\phi_{4,8*}$ - - - - - - - - - - - - -

$\phi_{5,5}\phi_{5,6}\phi_{5,7}\phi_{5,8*}$ - - - - - - - - - - - - -

$\phi_{6,6}\phi_{6,7}\phi_{6,8*}$ - - - - - - - - - - - - -

$\phi_{7,7}\phi_{7,8*}$ - - - - - - - - - - - - -

$\phi_{8,8*}$ - - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

- - - - - - - - - - - - -

ϕ = a→z; Ø→9 ; A→Z

It will be appreciated that such a matrix has an infinite possible number of entries. However, as indicated in the above representation of the matrix, it has been found sufficient to truncate the matrix and use only a very few of the potential PICs represented by such a matrix. For instance, in the above representation only those 48 entries actually shown are utilized for this exemplary embodiment. Furthermore, it should be noted that the last 7 entries of the last full column of entries in the above matrix has been truncated so as to, in effect, include an overlay of any potential entries to the right of that element in that particular row. For instance, the entry $\phi_{2,8+}$ indicates that this particular PIC represents the second character of any word having 8 or more characters therein.

Accordingly, in the exemplary embodiment utilizing only 48 different matrix entries, and where ϕ takes on any one of the 62 possible character values noted above, it follows that there are 2,976 PICs in all. (i.e., 48 X 62 = 2,976).

Thus, in the exemplary embodiment depicted by FIGURES 1 and 2, there would actually be 2,976 arrays wherein array No. 1 might correspond to PIC $a_{1,1}$; array No. 2 might correspond to PIC $a_{1,2}$; array No. 3 might correspond to PIC $a_{1,3}$; array No. 4 might correspond to PIC $a_{1,4}$; etc., on through the remaining 2,972 PICs. As should now be appreciated, in this exemplary embodiment, since there are 2,976 PICs in total, the retrieval file will comprise 2,976 possible binary bits for each record in the base data file.

Accordingly, if the base data file comprises 100,000 records, the retrieval file will, in turn, comprise 297.6 million bits of binary coded information. Of course, as should now be appreciated, one can selectively reduce the number of PICs in the retrieval file for a given application unless an unacceptable or undesirable loss in retrieval precision would accompany such a reduction or reorganization of PICs for the particular data base in question.

As should also be appreciated, the retrieval file might comprise machine readable data such as magnetically encoded on typical conventional magnetic disc drives, magnetic tapes, magnetic core storage arrays, etc. Furthermore the retrieval file may be both machine readable and humanly readable if it is stored in the form of arrays of photographic images, for example wherein each element is coded as the presence or absence of a transparent or non-transparent area in a film, etc.

In any event, once the retrieval file has been constructed, it may be utilized for accessing desired ones of the records in the base data file 100. For example, the input search data depicted at 106 might form still another (but much smaller) file, a search data file 108. Typically, the input search data would comprise a portion of a word, a whole word or a group of words which the user expects to be found within the record he desires to access.

Accordingly, the search data file may itself be processed in much the same manner as was the base data file

in the original construction of the retrieval file. That is, the search data file is analyzed with respect to the same set of PICs represented by the set of arrays in the retrieval file 104. Once the particular ones of these PICs present in the search data file have been identified, then the corresponding arrays from the retrieval file representing these particular PICs are selected and individual elements or binary values thereof in all of the selected arrays are compared with one another to identify the addresses of records in the base data file having all the desired PICs.

For example, as shown in FIGURE 2, the record stored at location or address 0304 is the only record having all of PICs 1 through 4. On the other hand, the record stored at location or address 0000 does not have PIC #3 but does have PICs 1, 2 and 4. As should now be apparent, the desired set of PICs identified by processing the input search data 106, 108 corresponds to a set of arrays in the retrieval file and by machine or otherwise comparing the binary values of corresponding elements in each array of the set, it may be easily determined that only records having certain now-identified addresses comply with the request or inquiry at 106. Of course, the inquiry information at 106 may well effectively request the absence of some particular PIC as well as the presence of some particular PIC, etc., as should be appreciated.

Accordingly, as indicated in FIGURE 1 at 110, the desired retrieval arrays are selected and compared to

identify addresses of desired records whereupon the accessible base data file 100 is accessed as indicated at 112 in FIGURE 1 to select the corresponding records and to display them as indicated at 114. Of course, the display may be in a form of a projection of a photographic image, a cathode ray tube or other electronic display of machine readable records, hard copy generated either from machine readable or photographic image records, etc.

Once the input search data 106 is presented, it is thus only necessary to identify the particular PICs corresponding thereto, to select the arrays corresponding to these PICs to compare the corresponding elements of the selected set of arrays thus identifying the location of desired records in the base data file and to then select and display the corresponding records.

If desired, all of these steps may be manually performed. Alternatively, some or all may be machine implemented. For instance, the previously referenced U.S. Patent No. 3,354,467 to Beekley teaches apparatus that can be utilized to automatically perform the comparison step of these search and retrieval processes.

In the preferred embodiment and preferred mode of the invention, the identification of desired PICs, selection and comparison of corresponding arrays and identification of the addresses of desired records is all automatically accomplished by a programmed general purpose data processor in conjunction with conventional input/output devices and digital information storage devices. Furthermore, in the preferred embodiment of the invention, the accessible base

data file or records 100 is itself a conventional storage device which may be automatically accessed by the programmed general purpose data processor such that the selection of a particular record at the identified address and its display are also automatically performed.

As a specific example of the above-described process for constructing the binary coded arrays in the retrieval file, FIGURES 3 and 4 have been prepared as a specific example showing the coding of the following text which is set forth here as an example of the text of any one of the records in the base data file 100 in FIGURE 1. This text may, of course, be of any given length but will, in the exemplary embodiment, always be encoded in binary form in the retrieval file depicting the presence or absence of each of the individual 2,976 PICs. For instance, as shown in FIGURE 3, only a relatively few of the 2,976 potential PICs are present in this exemplary text:

1 million ransom note Police press search for 2 men and missing wife of millionaire AP Minneapolis Minneapolis FBI Virginia Piper Harry C Jr. Orono Hennepin County Jaffray Hopwood brokerage investment firms kidnapers Lake Minnetonka Mrs George Partridge III Addison L Tad David Lewis Associates Senate votes to ban war use of rain making AP Washington burning of forests U S weapons Indochina amendment Sen Gaylor Nelson D Wis Defense Secretary Melvin R Laird North Vietnam cloud seeding Laos Cambodia Ho Chi Minh Trail weather modification activities House chemical sprays Firestorm operations reported Pentagon Southeast Asian jungles New York Times firestorm operations Sherwood Forest Hot Tip John Tower R Tex military helicopters Ft Sam Houston Tex Ft Lewis Wash Carson Colo Luke Air Force Base Ariz Mountain Home Idaho Work starts on gym Columbia University New York AP gymnasium campus Morningside Park Chicken exemption opposed WASHINGTON AP Cost Living Council price controls

Due to space considerations, and in an effort to show all 2,976 PICs, these entries have been shown in FIGURE 3 in two-dimensional form wherein the ϕ or alphabetic value of the character is represented by the ordinate and the character position in words having 1, 2, et. seq. total characters is represented by the abscissa.

I should be noted that this particular text example as coded in FIGURE 3 and 4 represents an extract from some full text record. Nevertheless, even this extract contains considerable redundancies which are irreversibly coded in the set of PICs shown in FIGURE 3. For instance, the exemplary text includes the words TAD, TIP and TEX. Since these are all three character words with the initial character of T, it follows that there is only a single binary representation for these initial letters of these three words in the PIC matrix as shown in FIGURE 3. Furthermore, since the word "SAM" is also included in this text, it is clear that the second letter A of SAM and the second letter A of TAD are likewise represented by but a single binary coded PIC in the matrix shown in FIGURE 3. Similar comments can be made for many of the entries as will become apparent by a study of the above quoted text with respect to the explicit binary coding of the PIC matrix shown in FIGURE 3.

Actually, the 2,976 PICs in the exemplary embodiment are not associated together as indicated for exemplary purposes in FIGURE 3. Rather, as indicated previously in FIGURES 1 and 2, the PICs correspond to individual arrays

of elements which elements, in turn, correspond to the address or location of particular records in the base data file. Accordingly, FIGURE 4 is simply a redrawing of FIGURE 3 showing a portion of several of the 2,976 arrays in the exemplary embodiment of FIGURES 1 and 2 that would result from the encoding of the above noted document or text.

For instance, assume that this particular record is stored in location 4104. Accordingly, as indicated in FIGURE 4, the array corresponding to PIC $A_{1,1}$ would contain a binary code indicating the absence of this particular PIC in the record located at address 4104. However, array corresponding to PIC $A_{1,2}$ would contain the other value of binary coding (indicated by cross-hatching in FIGURE 4) indicating the presence of this particular PIC in the document located at address 4104 (for instance, this coding would result because the word "AP" appears in the text). The other entries for the arrays shown in FIGURE 4 may be similarly confirmed by reference to the text and to FIGURE 3. Of course, there would be 2,936 additional arrays in the actual retrieval file of FIGURES 1 and 2.

It should also be noted at this point that the arrays of the retrieval file might not be disposed in planar organizations such as schematically indicated in FIGURES 1 and 2. In some embodiments, such a planar organization of arrays might enhance the comparison process for corresponding elements in selected arrays, etc. However, it should be particularly noted that it is not necessary to associate the arrays with planar organizations as indicated for

explanatory purposes in FIGURE 1 and 2.

In fact, in the preferred embodiment where the retrieval file comprises arrays stored as machine readable magnetically encoded data on conventional magnetic discs, etc., each array would in fact simply comprise a string of binary bits associated with one another in a conventional manner that might or might not involve any particular predetermined physical locations on the magnetic discs, etc.

As should now be appreciated, each word in the text of a record can be easily analyzed within conventional general purpose data computing equipment to detect its language structure such as the alphabetic value of each character and its position within the word and the number of characters in the word, etc., and thus defining the binary value of each of 2,976 PICs with respect to that given document. Accordingly, all of the binary coding shown in FIGURE 3 may be quickly and easily machine generated from the above quoted text corresponding thereto.

Thus, the construction step 102 shown in FIGURE 1 need not be manual but can comprise entirely machine processing once the appropriate information from the base data file is itself in machine readable format. Similarly, all of the records in the base data file 100 can be processed in this manner to construct all of the arrays in the retrieval file 104 as should now be apparent.

In a similar manner, any input words or portions thereof may be automatically analyzed to determine its

language structure such as the character values and their positional locations within the word of a particular size, etc., thus automatically identifying the particular PICs corresponding to the input search data. The arrays corresponding thereto may then be automatically selected by the programmed computer and the corresponding elements therein compared to identify the addresses of desired records.

Of course, all of these steps could also be carried out in an entirely manual operation and/or various degrees of automatic processing might be substituted for such manual operations. For instance, the arrays might be maintained as plates of film images which are manually semiautomatically extracted and compared with automatic photoelectric reading of the superimposed binary coded elements, etc.

For some applications, it may be preferable to decrease the number of PICs but to generate and maintain a plurality of retrieval files for such PICs. That is, a first retrieval file for the given set of PICs would correspond to a particular predetermined portion of the information on the base data file or stored records, while the second retrieval file would be coded to reflect the information content of another predetermined portion of such records, etc. For instance, if the base data file comprises telephone directory records, one might want to organize several retrieval files wherein one of the retrieval files relates to the name associated with each telephone

directory record and wherein another retrieval file relates to the street address associated with each telephone record, etc. However, as may also be appreciated, these separate retrieval files for a common data base may simply comprise separate groups of PICs within the same retrieval file. In short, the characteristic of language location or meaning within each record may be used in combination with alphabetic values, position within words, etc., as a part of the language structure represented by the PICs in the retrieval file.

That is, this may also be thought of as a single retrieval file wherein a first set of PICs includes the characteristic that it is associated with the name portion of each corresponding record in the base data file whereas another segment of PICs in the retrieval file includes the characteristic that it is associated with the street address portion of such records, etc. In any event, when a retrieval file or files are so organized, the input search data may also be organized accordingly and noted as relating to the name portion of such records or to the street portion of such records, etc., with the corresponding selection of arrays from the retrieval file being made from that particular portion of a retrieval file or a particular retrieval file, as appropriate, thus increasing the precision with which retrieval is accomplished.

If the entire text of each record in the base data file has been utilized in constructing the binary coded arrays in the retrieval file, then every word in the

text of each record is a "key word". However, these key words can be used as input queries in any given order, etc., without affecting the output results. Furthermore, the selection of key words is not a function of some intermediate interpreter who has beforehand selected only a few of the words from the text of the record for use as such "key words".

The retrieval file may be made more flexible by modifying the selection of PICs so as to, in effect, overlay smaller character groups on larger character groups in the coding process. In effect, this has already been done in the exemplary embodiment with respect to word lengths of eight or more characters as previously noted. However, it is also possible to carry the process even further, thus decreasing the total number of PICs required for the retrieval file and saving storage space, etc.

Of course, this will increase the coding ambiguities inherent in the retrieval file and thus decrease the precision of retrieval capability to some extent. However, it will at the same time increase the flexibility of the retrieval system since one is then capable of retrieving with input or inquiry words without knowing exactly how many characters are in the finding word.

That is, for instance, if the retrieval file is related to a base data file of telephone directory records, one may not know the exact spelling or the number of characters in a person's name on the particular record desired. However, it is usually possible to provide at

least the first few letters of the name for the record being requested, and, if the retrieval file has been coded so as to in effect overlay small character groups on larger character groups, then one may successfully retrieve the desired information even though the total number of characters in the name in question is unknown. Of course, since the coding ambiguities in such a retrieval file are necessarily increased to provide the increased flexibility, the precision with which retrieval may be effected is correspondingly decreased. Accordingly, in addition to the desired name, a retrieval using such a retrieval file may often produce retrieval results calling forth records other than the particular one actually desired. Nevertheless, the accuracy of the recall is still 100 percent in that the desired record will surely be among those that are requested and provided by the retrieval system.

The preferred embodiment involves a retrieval file wherein the PICs are chosen to represent the alphabetic value and relative sequential location of characters in groups of characters such as words contained on record in the base data file. However, it should now be apparent that the selection of PICs can be related to the language structure of the base data file records in other manners as well. For example, if the base data file contains records having many chemical or mathematical formulae, etc., some of the PICs may represent mathematical symbols, operations, quantities, etc. Even particular complete words might be designated as a PIC in a particular retrieval file for a particular

application related to some particular base data file of records, for which such a PIC selection would be advantageous. For instance, if the base data file comprises clippings from a newspaper, one of the PICs in the retrieval file might well represent the presence or absence of an accompanying picture associated with the text on a given record, etc. Another PIC might relate to whether or not the record in question is an obituary or whether or not the particular record in question was written by a particular columnist, etc. In short, depending upon the type of base data file involved and the responsiveness desired by the user, one may adapt the set of PICs in a particular retrieval file to the particular language structure of the base data file records in many different ways to achieve particular desired end results as should now be apparent.

One exemplary embodiment of apparatus for practicing this invention is shown in FIGURE 5 as comprising a programmed computer 120 together with associated peripheral equipment such as magnetic disc drive and storage units 122, 124; a cathode ray tube display and keyboard input/output unit 126 and possibly a paper tape punch and reader input/output unit 128 and/or other peripheral (not shown) such as magnetic tape drives. The paper tape or magnetic tape peripherals may be utilized, for instance, in inputting control program(s) to the magnetic core of computer 120 and thus conditioning it or adapting it for operation according to this invention.

In this particular embodiment, the accessed base data file of records and the retrieval file of binary coded arrays are both stored on the magnetic disc units 122, 124 and/or further magnetic disc storage units as required. The CRT display and keyboard unit 126 is then used for inputting the search data or inquiry information and for displaying the retrieved record as a result of such inquiries.

Another exemplary embodiment of equipment for practicing this invention is shown in FIGURE 6. Again, a programmed computer 130 is associated with the magnetic disc device 132 and a CRT display and keyboard input-output device 134. In addition, programmed computer 130 interfaces with a film image storage and retrieval device 136, which is adapted to store optical images of a set of information bearing records at predetermined addresses and is adapted for automatically delivering such information (for example at a display unit 138) corresponding to the information content of any given record when provided with the address of that given record by a computer 130. Such devices are available, for instance, to handle microfiche images from companies such as Image Systems at Culver City, California or Remington Rand. In this exemplary system, the base data file of records in accessible form would be stored at 136 while the retrieval file of binary coded arrays would be stored on the magnetic disc unit(s) 132. Inquiry information would then be input from the CRT display and keyboard unit 134. The programmed computer 130 would then analyze the inquiry information, identify the PICs represented thereby; select and compare the corresponding retrieval

arrays from the retrieval file to identify the addresses of the desired records and provide that address information to the storage and retrieval device 136 whereupon the requested records from the identified addresses would be displayed at 138. Of course, the display 138 might include copying means, etc., as appropriate and/or as desired.

TELEPHONE DIRECTORY ASSISTANCE RETRIEVAL SYSTEM

A. In General.

This section describes in detail a specific exemplary implementation of the invention that has actually been experimentally used in retrieving existing machine readable telephone directory records to provide an efficient telephone directory assistance service.

The apparatus for this embodiment corresponds to that shown in FIGURE 5 with the addition of three magnetic tape drives interfaced with computer 120 for initial processing of the telephone directory records since the information happened to be readily available in machine readable form on magnetic tapes.

In this exemplary embodiment, computer 120 is a Model PDP 8/E computer available from Digital Equipment Corporation, Maynard, Massachusetts. Magnetic disks 122, 124 comprise a Model DF-32 D disk file (32 K words) and control therefor also available from Digital Equipment Corporation and DD 140/2 disk drives (2314 type) and control therefor available from Diva, Inc., Eatontown, New Jersey. The CRT display and keyboard 126 comprises an S4300 Model CRT display unit available from Ontel Corporation, Plainview, New York. Paper tape punch and reader 128 is a PC 8-E Model paper tape reader/punch and control therefore available from Digital Equipment Corporation. In addition, a model 1045 NRZI magnetic tape transport (a track, 800 bpi) and a Model 1045 PE magnetic tape transport (9 track, 1600 bpi) both available from Wangco, Santa Monica, California are also provided together

with a Model 5091-P8 magnetic tape controller available from Datum, Inc., Anaheim, California. Of course, all controls have interface to the PDP 8/E mini-computer.

The structure and functioning of this computer equipment, per se, should already be apparent to those in the art. If not, reference may be had to "Small Computer Handbook", published by Digital Equipment Corporation, Maynard, Massachusetts and to "S4200 (4300) User Manual" published by Ontel Corporation, Plainview, New Jersey.

The computer 120 is a "mini" computer and is specially adapted to cooperate with the above-noted peripherals according to this invention by programming it as explained below. The programming is in the standard assembly level programming language recommended by the manufacturer and explained in detail in published literature such as "Introduction To Programming", published by Digital Equipment Corporation, Maynard, Massachusetts; "Software Manual, Magnetic Tape Controller", publication No. 1250.0 by Datum, Inc., Anaheim, California; and "Programming Manual For Diva Disc Systems Used With PDP 8/E" published by Diva, Inc., Eatontown, New Jersey.

The programming or "software" used in this exemplary embodiment is broadly indicated in schematic form at FIGURE 7. Since the telephone directory records are, in this instance, already in machine readable form on magnetic tapes 140, this data is first converted to a

standard format at 142 before the accessible base data file 100 and retrieval file 104 are constructed at 144 (including the step shown at 102 in FIGURE 1) and stored on magnetic disc (122, 124 in FIGURE 5).

As will be appreciated, the conversion 142 and construction 144 shown in FIGURE 7 are accomplished once in the start-up of the whole information storage and retrieval system and thereafter only as necessary to take into account changes in the base data file. Such changes may be accommodated by a complete reconstruction process based on a whole new input file of records 140 or as a special update modification of the accessible base data file and retrieval file already stored on magnetic disc. Since suitable updating programs for file maintenance purposes are well known, per se, in the art, and since same are actually not necessary to practice this invention, no detailed description of such updating programs will be given.

Once the retrieval file and base data files are in existence, then the system is ready for retrieval processing as shown at 146 in FIGURE 7. Here, inquiry inputs from the CRT display and keyboard unit 126 are accepted and retrieved telephone directory data are shortly thereafter displayed thereon.

These three basic areas of this exemplary embodiment will now be explained in more detail.

B. Conversion.

B1. In General.

The conversion 142 is shown more explicitly in

FIGURE 8. It comprises two programs and merely serves to enter all relevant information into the system in a standardized format from pre-existing available data on magnetic tape.

The pre-existing data base is magnetic tape oriented and comprises two files: listings 150 and captions 152. Both are processed by Program No. 1 to produce standardized files 154, 156 having standard formatting, coding, etc. These standardized tapes are then merged by Program No. 2 into a final file 158 which is formatted for use by the retrieval and base data file construction program segment 144.

To increase the flexibility and speed of the final retrieval process, the standardized file 158 is actually separated into alphabetic segments, each of which is to be considered as a separate base data file 100 and for each of which a retrieval file(s) 104 will be constructed. The alphabetic segments are chosen so as to result in roughly equal sized base data files. In the exemplary embodiment, 16 alpha groupings or segments were chosen:

16 Alpha Groupings

1. E - L
2. Mi - N
3. Br through Bz - W
4. B through Bq - Cr through Cz - Q
5. Z - I - P
6. Ca through Cq
7. U - J - K
8. V - R
9. D - Y - X
10. G
11. H
12. F
13. S through Sn
14. T - A
15. O - So through Sz
16. M - Mh

B2. Description of Input Magnetic Tape Files 150,152

1. Physical Characteristics: 9 track, 800 bpi, regular mode.

2. Structure: Variable length blocks, variable length records per block, maximum characters per block-2048. Record terminator (477_8). Block terminator (75_8).

3. Record Structure:

A. Fixed Length Control Field (58 characters length)

0: N/A
1: field code (47_8)
2-4: N/A
5-7: NPA (exchange)
8: Borough Code
9-15: Telephone #
37: listing type
38-57: N/A

B. Full name field- 0: field code

C. Full title- 0: field code

D. Full desig.- 0: field code

E. Supp Sort Criteria (36 characters length)

0: field code
1-29: N/A
30-36: Tel # or Z for non pub

F. Caption control # (N/A)- 0: field del

G. Listing name

H. Listing title

I. Listing desig.

J. Listing street

K. Listing house #

L. Listing house # suffix

M. Listing locality

O. Remainder of fields: N/A

4. Character Representation

23_8-34_8: characters 0-9

40_8-71_8: characters A-Z

Note: that these characters may also include parity bit

B3. Description of Standardized Magnetic Tape Files 154, 156, 158

1. Physical Characteristics: 9 track, 1600 bpi, special core dump mode.

2. Structure: Fixed length block (512 words), one record per block.

3. Record Structure:

A. Header (16 words)

1. Locations 2,3,4-double precision sequence count, mpf (N/A).

2. Location 11- non pub info (YES=4000; NO=0)

3. Location 12-listing type (Bus=20; Prof=10; Res=0)

4. Other locations- N/A

B. NPA (area code)- 3 words

C. Telephone #-5

D. Borough- 2 1/2

E. Full Name- 73 1/2

F. Full Title- 7 1/2

G. Full Designation- 19 1/2

H. Listing Name- 67 1/2

I. Listing Title- 7 1/2

J. Listing Designation- 19 1/2

K. Street- 17 1/2

L. House #- 5

M. House # Suffix- 3 1/2

N. Locality- 16 1/2

O. Blank- 248

4. Standardized Coding Structure

Character	Octal	Character	Octal
A	33	0	65
B	34	1	66
C	35	2	67
D	36	3	70
E	37	4	71
F	40	5	72
G	41	6	73
H	42	7	74
I	43	8	75
J	44	9	76
K	45	Space	77
L	46	@	0-77*
M	47	(	0-50*
N	50	)	0-51*
O	51	"	0-75*
P	52	:	0-72*
Q	53	$	0-44*
R	54	%	0-45*
S	55	;	0-73*
T	56	&	0-46*
U	57	'	0-47*
V	60	-	0-55*
W	61	*	0-52*
X	62	.	0-56*
Y	63	/	0-57*
Z	64	#	0-43*
		,	0-54*
		field delimiter	0-41*

* These characters are double coded.
Note: Blanks are double zeroes.

B5. Functional Description of Conversion Programs

1. Program #1

A. Obtain relevant elements from the input tape data base

1. NPA (area code)
2. Borough
3. Telephone number; non-pub info.
4. Full name
5. Full title
6. Full designation
7. Listing name
8. Listing title
9. Listing designation
10. Street
11. House number
12. House number suffix
13. Locality
14. Listing sequence
15. Non-pub info.
16. Listing type

B. Format to Standardized Specifications

1. Magnetic tape output; fixed length (512) blocks; one record per block; 9 track; 1600 bpi; special core dump mode.
2. Standardized codes

2. Program #2

A. Merge the standardized captions and listings files.

B. Add blank records for editing maneuverability.

C. Split into 16 separate and distinct alpha groups.

B6. Detailed Description of Program No. 1

Program No. 1 is shown in block form at FIGURE 9 with subroutines utilized therein detailed in FIGURES 10-17. An explicit listing of the assembly level source program language for Program No. 1 and all related subroutines follows. With respect to FIGURE 9, it will be noted that program sections I-V correspond to specific listing statement numbers:

Section	Instruction Statements
I	0200- 0213
II	0214- 0216
III	0217- 0260
IV	0261
V	0262- 0273

These five program sections are functionally described below as an introduction to the explicit source program listing:

I. Initialize

A. OH, LH are a double precision counter. This counter is used to allow a maximum 24,576 records to be output on a magnetic tape in the initial standardized file.

B. SEQH, SEQL are a double precision record sequence counter. It is stored in each record in position 2 and 3.

C. REOF is a single precision counter. It is used to allow a normal return from XREAD upon encountering the first two tape marks (after file and header labels). XREAD concludes tape processing upon encountering the third.

D. Four XREAD commands are dummy commands. They essentially pass the tape over the header and file labels and their associated tape marks.

II. XREAD: Input tape block. Inputs of the input tape file. If REOF signals that the third tape mark has been reached, the input tape is rewound and the proram halts at location 416, otherwise normal return. XREAD merely sets up the parameters for the magnetic tape operation, sends control to XMAG which accomplishes the physical magnetic tape operation, and received control from XMAG at completion and acceptance of the operation. XMAG also delineates tape mark vs normal record.

III. Standardized processing

A. XBUF: pads standardized record before data entry.

1. Pads data portion with spaces

2. Pads remainder with zeroes

B. Sequence # and mpf put into standardized header (locations 2, 3, 4).

C. The desired elements are retrieved from the input record by using the GETFLD and DOFLD subroutines. GETFLD merely positions the input data pointer at the beginning of the next field encountered. DOFLD, assuming the data pointer is at the beginning of the data field, goes to a list to get the number of characters desired transferred and the location of the DRC field to which the NYT field will be transferred. Exit from the subroutine is made upon encountering the next field delimiter in the input

data. Also, at the time of transfer, the codes are changed to the standardized code (via PUTCHR) which is a 6 bit code (except for double codes).

IV. XWRITE: Output standardized record. Uses OH and OL as output counters. Upon overflow, the output tape is given a tape mark and rewound and the program halts at location 435. Note that XMAG is the subroutine which actually accomplishes physical magnetic tape operations.

V. Search and Distinguish Terminator. The remainder of the input record is scanned until reaching either an end of block code (75_8) or an end of record code (477_8). End of record causes the program to go to Section III of the mainline, while end of block causes the program to go to Section II.

Program No. 1

```
BSY=7702
MQ1=7421
MQ0=7501
SWP=7521
CDF0=6201
CDF1=6211
CDF2=6221
DMCR=6517
DCBRH=6501
DCBRL=6500
DSSRH=6503
DSSRL=6502
DESR0=6511
DESR1=6515
DWCR=6516
DDMAR=6514
DCSR=6512
DAIRL=6504
DAIRH=6505
DUSRL=6506
DUSRH=6507
GTF=6004
RTF=6005
RMF=6244
RIB=6234
SRQ=6003

FIXFAB
        *=0
```

```
0020    0600    XMAG,       MAG
0021    0442    XWRITE,     WRITE
0022    0400    XREAD,      READ
0023    0456    XBUF,       BUF
0024    0000    SEQL,       0
0025    0000    SEQH,       0
0026    3003    FL,         3003
0027    3002    FH,         3002
0030    3004    MPF,        3004
0031    0000    CNT,        0
0032    0000    OTLOC,      0
0033    0000    TMP,        0
0034    0000    REOF,       0
0035    0000    OH,         0
0036    0000    OL,         0

0037    1177    C1177,      1177
0040    7703    CM75,       -75
0041    7376    CM402,      -402
0042    7304    CM474,      -474
0043    0041    C41,        41
0044    1000    C1000,      1000
0045    3777    C3777,      3777
0046    7700    C7700,      7700
0047    7772    CM6,        -6
0050    0077    C77,        77
0051    4001    C4001,      4001
0052    7776    CM2,        -2

                PAGE

0200    7300    START,      CLA CLL
0201    1047                TAD CM6
0202    3035                DCA OH
0203    3036                DCA OL
0204    3025                DCA SEQH
0205    3024                DCA SEQL
0206    7346    NYTST,      CLA CLL CMA RTL  /SET AC = -3
0207    3034                DCA REOF
0210    4422                JMS I XREAD
0211    4422                JMS I XREAD
0212    4422                JMS I XREAD
0213    4422                JMS I XREAD      /BYPASS LABELS

0214    4422    DOBLK,      JMS I XREAD
0215    7240                CLA CMA
0216    3010                DCA 10   /SET INPUT BUF
```

```
0217  4423   DOREC,    JMS I XBUF        /INIT OUTPUT BUF
0220  2024             ISZ SEQL
0221  5223             JMP .+2
0222  2025             ISZ SEQH
0223  1024             TAD SEQL
0224  3426             DCA I FL
0225  1025             TAD SEQH
0226  3427             DCA I FH
0227  7201             CLA IAC
0230  3430             DCA I MPF
0231  1037             TAD C1177
0232  3011             DCA 11
0233  4274             JMS GETFLD        /GO TO FIELD 1
0234  4776             JMS I XTYPE
0235  2010             ISZ 10
0236  2010             ISZ 10
0237  2010             ISZ 10   /BYPASS 1ST 3 CHRS
0240  4304             JMS DOFLD         /PROCESS EXCHANGE (NPA)
0241  4304             JMS DOFLD         /       "        BOROUGH
0242  4304             JMS DOFLD         /       "        TEL #
0243  4274             JMS GETFLD        /GO TO FIELD 2
0244  4304             JMS DOFLD         PROCESS FULL NAME
0245  4304             JMS DOFLD         /       "       "     TITLE
0246  4304             JMS DOFLD         /       "       "     DESIGNATION
0247  4775             JMS I XNP         /PROCESS NON PUB
0250  4274             JMS GETFLD
0251  4274             JMS GETFLD
0252  4304             JMS DOFLD         /PROCESS LISTING NAMES
0253  4304             JMS DOFLD         /       "       "     TITLE
0254  4304             JMS DOFLD         /       "       "     DESIGNATION
0255  4304             JMS DOFLD         /       "       "     STREET
0256  4304             JMS DOFLD         /       "       "     HOUSE
0257  4304             JMS DOFLD         /       "       "     HOUSE # SUFX
0260  4304             JMS DOFLD         /       "       "     LOCALITY
0261  4421             JMS I XWRITE      /OUTPUT DRC RECORD

0262  6211   NKCR,     CDFI
0263  1410             TAD I 10
0264  6201             CDFO
0265  1040             TAD CM75
0266  7450             SNA       /END OF BLOCK?
0267  5214             JMP DOBLK         /YES
0270  1041             TAD CM402
0271  7650             SNA CLA   /END OF RECORD
0272  5217             JMP DOREC         /YES
0273  5262             JMP NKCR

0274  0000   GETFLD,   0
0275  6211             CDFI
0276  1410             TAD I 10
0277  6201             CDFO
0300  1042             TAD CM474
0301  7650             SNA CLA   /FIELD DEL?
0302  5674             JMP I GETFLD      /YES
0303  5275             JMP GETFLD+1
```

```
0304  0000  DOFLD,   0
0305  1411          TAD I 11
0306  3031          DCA CNT
0307  1411          TAD I 11
0310  3032          DCA OTLOC
0311  4344          JMS PUTCHR      /DRC FIELD DEL = +41
0312  1043          TAD C41
0313  4344          JMS PUTCHR
0314  6211  NXCRT,   DCF1
0315  1410          TAD I 10
0316  6201          CDF0
0317  3033          DCA TMP
0320  1033          TAD TMP
0321  1042          TAD CM474
0322  7650          SNA CLA  /FIELD DEL?
0323  5704          JMP I DOFLD     /YES
0324  1033          TAD TMP
0325  0050          AND C77
0326  1044          TAD C1000
0327  3033          DCA TMP
0330  1433          TAD I TMP       /DRC CHAR
0331  7510          SPA      /SPEC CHAR?
0332  5337          JMP D2   /YES
0333  4344  D3,     JMS PUTCHR
0334  2031          ISZ CNT  /FIELD DONE?
0335  5314          JMP NXCRT       /NO
0336  5704          JMP I DOFLD
0337  0045  D2,     AND C3777
0340  3033          DCA TMP
0341  4344          JMS PUTCHR      /SPEC CHR = +CHAR
0342  1033          TAD TMP
0343  5333          JMP D3

0344  0000  PUTCHR,  0
0345  0050          AND C77
0346  7421          MQL
0347  1032          TAD OTLOC
0350  7500          SMA
0351  5364          JMP P2
0352  0045          AND C3777
0353  3032          DCA OTLOC
0354  7501          MQA
0355  7002          BSW
0356  7421          MQL
0357  1432          TAD I OTLOC
0360  0050          AND C77
0361  7501          MQA
0362  3432          DCA I OTLOC
0363  5744          JMP I PUTCHR

0364  7300  P2,     CLA CLL
0365  1432          TAD I OTLOC
0366  0046          AND C7700
0367  7501          MQA
0370  3432          DCA I OTLOC
0371  1032          TAD OTLOC
0372  1051          TAD C4001
0373  3032          DCA OTLOC
0374  5744          JMP I PUTCHR

0375  0504  XNP,     NP
0376  0527  XTYPE,   TYPE

      PAGE
```

```
0400    0000    READ,       0
0401    4420                JMS I XMAG
0402    5020                5020        /COM
0403    0000                0           /ADR
0404    0000                0           /WC
0405    0001                1           /EXT REG
0406    2034                ISZ REOF            /EOF
0407    5600                JMP I READ          /OK
0410    4420                JMS I XMAG
0411    5010                5010
0412    0000                0
0413    0000                0
0414    0000                0
0415    7000                NOP
0416    7402                HLT
0417    5620                JMP I XNYT
0420    0206    XNYT,       NYTST
0421    4420    TEND,       JMS I XMAG
0422    0050                50
0423    0000                0
0424    0000                0
0425    0000                0
0426    7000                NOP
0427    4420                JMS I XMAG
0430    0100                10
0431    0000                0
0432    0000                0
0433    0000                0
0434    7000                NOP
0435    7402                HLT
0436    1047                TAD CM6
0437    3035                DCA OH
0440    3036                DCA OL
0441    5642                JMP I WRITE

0442    0000    WRITE,      0
0443    4420                JMS I XMAG
0444    0040                40          /COM
0445    3000                3000        /ADR
0446    7000                -1000       /WC
0447    0100                100         /EXT REG
0450    7402                HLT         /EOF
0451    2036                ISZ OL
0452    5642                JMP I WRITE
0453    2035                ISZ OH
0454    5642                JMP I WRITE
0455    5221                JMP TEND

0456    0000    BUF,        0
0457    7300                CLA CLL
0460    1303                TAD C2777
0461    3012                DCA 12
0462    1300                TAD CM1000
0463    3031                DCA CNT
0464    3412    NXB1,       DCA I 12
0465    2031                ISZ CNT
0466    5264                JMP NXB1
0467    1302                TAD C3017
0470    3012                DCA 12
0471    1301                TAD CM420
0472    3031                DCA CNT
0473    7240    NXB2,       CLA CMA
0474    3412                DCA I 12
0475    2031                ISZ CNT
0476    5273                JMP NXB2
0477    5656                JMP I BUF
```

```
0500  7000  CM1000,   -1000
0501  7360  CM420,    -420
0502  3017  C3017,    3017
0503  2777  C2777,    2777


0504  0000  NP,       0
0505  1010            TAD 10
0506  1341            TAD NC47
0507  3351            DCA POS
0510  6211            CDF1
0511  1751            TAD I POS
0512  1042            TAD CM474
0513  7640            SZA CLA          /AT EXCHANGE?
0514  7402            HLT              /NO
0515  1010            TAD 10
0516  1342            TAD NC34
0517  3351            DCA POS
0520  1751            TAD I POS
0521  1346            TAD NCM471
0522  7650            CNA CLA          /Z?
0523  1347            TAD NC4000       /YES
0524  6201            CDF0
0525  3750            DCA I NC3011
0526  5704            JMP I NP
0527  0000  TYPE,      0
0530  1010            TAD 10
0531  1343            TAD NC44
0532  3351            DCA POS          /LOCATION OF LISTING TYPE
0533  6211            CDF1
0534  1751            TAD I POS        /LISTING TYPE
0535  6201            CDF0
0536  0344            AND NC30
0537  3745            DCA I L3012      /STORE IN DRC HEADER
0540  5727            JMP I TYPE

0541  0047  NC47,      47
0542  0034  NC34,      34
0543  0044  NC44,      44
0544  0030  NC30,      30
0545  3012  L3012,     3012
0546  7307  NCM471,    -471
0547  4000  NC4000,    4000
0550  3011  NC3011,    3011
0551  0000  POS,       0

            PAGE
            /FORMAT:          JMS I XMAG
            /           COMMAND
            /           ADDRESS
            /           WORD COUNT
            /           EXTENSION REGISTER
            /           RETURN:  EOF
            /           RETURN:  NORMAL
```

```
0600    0000    MAG,        0
0601    1337                TAD MCM12
0602    3351                DCA REVCNT        /SET 10 RETRIES
0603    1600                TAD I MAG
0604    2200                ISZ MAG
0605    3352                DCA COM/COMMAND
0606    7240                CLA CMA
0607    1600                TAD I MAG
0610    2200                ISZ MAG
0611    3353                DCA CADDR         /GET CURRENT ADDRESS
0612    1600                TAD I MAG
0613    2200                ISZ MAG
0614    3354                DCA WRDCNT        /GET WORD COUNT
0615    1600                TAD I MAG
0616    2200                ISZ MAG
0617    3355                DCA COMEX         /GET EXT REGISTER
0620    3350                DCA MADCOM
0621    1352    RETRY,      TAD COM
0622    1350                TAD MADCOM
0623    4322                JMS SETCOM        /SET CONTROLLER FOR FUNCTION
0624    1354                TAD WRDCNT
0625    3032                DCA 32
0626    1353                TAD CADDR
0627    3033                DCA 33    /SET WORD COUNT & CURRENT ADDRESS
0630    1355                TAD COMEX
0631    6717                6717
0632    7300                CLA CLL  /SET EXT REG
0633    4332                JMS MAGOP         /PERFORM MAGTAPE FUNCTION
0634    6706                6706
0635    7421                MQL       /STORE STATUS IN MQ
0636    7601                MQA
0637    0340                ANS MC6774
0640    7450                SNA
0641    5271                JMP MOK   /NO ERRORS
0642    0341                AND MC7677
0643    7450                SNA
0644    5273                JMP MEOF          EOF
0645    0342                AND MC3543
0646    7640                SZA CLA
0647    7402                HLT       /BAD REC OR OFFLINE
0650    2351    PAR,        ISZ REVCNT        PARITY
0651    5253                JMP .+2  /RETRY
0652    7402                HLT       /RETRY FAILURE
0653    1352                TAD COM
0654    0345                AND MC70
0655    1346                TAD MCM40
0656    7650                SNA CLA  /FAILURE ON MT?
0657    1343                TAD MC100         /YES
0660    3350                DCA MADCOM
0661    1352                TAD COM
0662    0344                AND MC7000
0663    1345                TAD MC70
0664    4322                JMS SETCOM        /SET CONTROLLER FOR BACKSPACE
0665    7240                CLA CMA
0666    3032                DCA 32   /WC = RECS BACKSPACED
0667    4332                JMS MAGOP         /PERFORM BACKSPACE
0670    5221                JMP RETRY         /AND TRY AGAIN
0671    2200    MOK,        ISZ MAG
0672    5600                JMP I MAG         /NORMAL EXIT
```

```
0674  0345                  AND MC70
0675  1346                  TAD MCM40
0676  7650                  SNA CLA  /EOF ON WRITE?
0677  5250                  JMP PAR  /YES - PARITY PROBLEM
0700  7000                  NOP      /REWIND UNIT
0701  5600                  JMP I MAG        /EXIT EOF
0702  0000  REWIND,         0
0703  1352                  TAD COM
0704  0344                  AND MC7000
0705  1347                  TAD MC10
0706  4322                  JMS SETCOM       /SET CONTROLLER FOR REWIND
0707  6722                  6722     /EXECUTE REWIND
0710  1352                  TAC COM
0711  0344                  AND MC7000
0712  7106                  CLL RTL
0713  7006                  RTL      /LOAD CON W/ NEXT MAG OF
0714  1357                  TAD NEXM
0715  3356                  DCA NEXCOM
0716  1756                  TAD I NEXCOM
0717  6716                  6716
0720  7300                  CLA CLL
0721  5702                  JMP I REWIND
0722  0000  SETCOM,         0
0723  6711                  6711
0724  5323                  JMP .-1
0725  6716                  6716
0726  6721                  6721
0727  5326                  JMP .-1
0730  7300                  CLA CLL
0731  5722                  JMP I SETCOM
0732  0000  MAGOP,          0
0733  6722                  6722
0734  6701                  6701
0735  5334                  JMP .-1
0736  5732                  JMP I MAGOP
0737  7766  MCM12,          -12
0740  2764  MC6774,         2764
0741  7667  MC7677,         7667
0742  3543  MC3543,         3543
0743  0100  MC100,          100
0744  7000  MC7000,         7000
0745  0070  MC70,           70
0746  7740  MCM40,          -40
0747  0010  MC10,           10
0750  0000  MADCOM,         0
0751  0000  REVCNT,         0
0752  0000  COM,            0
0753  0000  CADDR,          0
0754  0000  WRDCNT,         0
0755  0000  COMEX,          0
0756  0000  NEXCOM,         0
0757  0760  NEXM,           NEXTM
0760  1020  NEXTM,          1020     /0
0761  0040                  40       /1
0762  0040                  40       /2
0763  0040                  40       /3
```

1000	4077	4077
1001	0077	77
1002	0077	77
1003	4050	4050
1004	4051	4051
1005	4075	4075
1006	4072	4072
1007	4044	4044
1010	4045	4045
1011	4073	4073
1012	4040	4040
1013	4047	4047
1014	4055	4055
1015	4052	4052
1016	4056	4056
1017	0077	77
1020	0077	77
1021	0077	77
1022	4057	4057
1023	0065	65
1024	0066	66
1025	0067	67
1026	0070	70
1027	0071	71
1030	0072	72
1031	0073	73
1032	0074	74
1033	0075	75
1034	0076	76
1035	4054	4054
1036	4043	4043
1037	0077	77
1040	0033	33
1041	0034	34
1042	0035	35
1043	0036	36
1044	0037	37
1045	0040	40
1046	0041	41
1047	0042	42
1050	0043	43
1051	0044	44
1052	0045	45
1053	0046	46
1054	0047	47
1055	0050	50
1056	0051	51
1057	0052	52
1060	0053	53
1061	0054	54
1062	0055	55
1063	0056	56
1064	0057	57
1065	0060	60
1066	0061	61
1067	0062	62
1070	0063	63
1071	0064	64
1072	0077	77
1073	0077	77
1074	0077	77
1075	0077	77
1076	0077	77
1077	0077	77

```
1200    7775    -3
1201    7020    7020

1202    7777    -1
1203    7030    7030

1204    7771    -7
1205    7023    7023

1206    7557    -221
1207    3032    3032

1210    7763    -15
1211    7144    7144

1212    7733    -45
1213    3153    3153

1214    7573    -205
1215    7177    7177

1216    7763    -15
1217    3302    3302

1220    7733    -45
1221    7312    7312

1222    7737    -41
1223    3335    3335

1224    7770    -10
1225    7357    7357

1226    7773    -5
1227    7364    7364

1230    7737    -41
1231    3367    3367

BUF     0456
CADDR   0753
CM1000  0500
CM2     0052
CM402   0041
CM420   0501
CM474   0042
CM6     0047
CM75    0040
CNT     0031
COM     0752
COMEX   0755
C1000   0044
C1177   0037
C2777   0503
C3017   0502
C3777   0045
C4001   0051
C41     0043
C77     0050
C7700   0046
DOBLK   0214
DOFLD   0304
```

DOREC	0217
D2	0337
D3	0333
FH	0027
FL	0026
GETFLD	0274
L3012	0545
MADCOM	0750
MAG	0600
MAGOP	0732
MCM12	0737
CM40	0746
MC10	0747
MC100	0743
MC3543	0742
MC6774	0740
MC70	0745
MC7000	0744
MC7677	0741
MEOF	0673
MOK	0671
MPF	0030
NCM471	0546
NC30	0544
NC34	0542
NC4000	0547
NC44	0543
NC47	0541
NEXCOM	0756
NEXM	0757
NEXTM	0760
NL3011	0550
NP	0504
NXB1	0464
NXB2	0473
NXCR	0262
NXCRT	0314
NYTST	0206
OH	0035
OL	0036
OTLOC	0032
PAR	0650
POS	0551
PUTCHR	0344
P2	0364
READ	0400
REOF	0034
RETRY	0621
REVCNT	0751
REWIND	0702
SEQH	0025
SEQL	0024
SETCOM	0722
START	0200
TEND	0421
TMP	0033
TYPE	0527
WRDCNT	0754
WRITE	0442
XBUF	0023
XMAG	0020
XNP	0375
XNYT	0420
XREAD	0022
XTYPE	0376
XWRITE	0021

Operating Instructions for the above listed Program No. 1 are as follows:

1. Set up and load magnetic tapes. Load program into memory bank 0.

   input file-tape unit 1

   scratch-tape unit 0

2. SR=200 : load, clear, continue

Halts: 416- Input tape done.

A. To continue- put on next input tape and hit continue.

B. To terminate- set SR=421: load, clear, continue.

435- Output tape at limit: Put on next tape and hit continue.

647- Magtape error: If bit 6 in MQ indicates end of tape set SR=671 load, clear, continue.

652- Retry failure.

B7. Detailed Description of Program No. 2

Program No. 2 is shown in block form at FIGURE 18 with subroutines utilized therein detailed in FIGURES 19-24. An explicit listing of the assembly level source program language for Program No. 2 and all related subroutines follows.

This program is written in three different versions (herein called "loaders") to distinguish processing of those alphabetic groupings or segments comprising whole character groups from split character groups.

Loader 1 processes whole character groups. Loader 2 processes the first half of split character groups while loader 3 processes the second half of split character groups.

For example, the letter "A" file can be processed in its entirety, therefore loader 1 would be used. The letter "M", which has more listings than can fit in a single XM block, must be split. Thus Ma-Mh is the 1st half segment and utilizes loader 2, and Mi-Mz is the 2nd half segment and utilizes loader 3. In the split programs (2 & 3) the split defining character (2nd character of the full name field which starts the 2nd half segment-i for Mi-Mz) must be manually entered into the program into location 317 before processing is initiated.

Note that the major difference between the programs is how the "COMP" subroutine handles the search.

Referring to FIGURE 18, it should be noted that program sections I-V correspond to specific listing statements as follows:

Loader	Section	Instruction Statements
1	I	0340-0344
1	II	0200-0204
1	III	0205-0217
1	IV	0240-0246
1	V	0300-0310
2	I	0340-0344
2	II	0200-0204
2	III	0205-0217
2	IV	0240-0246
2	V	0300-0310

Loader	Section	Instruction Statements
3	I	0340-0344
3	II	0200-0204
3	III	0205-0217
3	IV	0240-0251
3	V	0300-0310

These five program sections are functionally similar if not identical and are functionally described below as an introduction to the explicit source program listing for loaders 1, 2 and 3 of Program No. 2.

I. Get the appropriate # of blanks from the SR (switch register). Note that this number is set up to be a multiple of 12 for simpler processing during the retrieval file storage. The actual number of blanks is determined manually with regard for the frequency of activity of the particular alpha segment.

II. Get the character to define the alpha group from the SR. Note that the split defining character is entered via toggle switch before processing.

III. Set the proper tape unit in the read command (listing or captions). Search through the file until the desired alpha segment is found, then transfer all subsequent records to the output file until a new alpha segment is encountered.

IV. If the listings file is currently being processed, change the tape unit to the captions file and repeat part III. If the captions file is currently being processed go to part V.

v. After both listings and captions files have been processed, blank records are added to each alpha segment per entry in Part I. Upon completion of this operation, the program is terminated.

Program No. 2, Loader 1

```
BSW=7002
MQL=7421
MQA=7501
SWP=7521
CDF0=6201
CDF1=6211
CDF2=6221
DMCR=6517
DCSRH=6501
DCSRL=6500
DSSRH=6503
DSSRL=6502
DESRO=6511
DESRI=6515
DWCR=6516
DDMAR=6514
DCSR=6512
DAIRL=6504
DAIRH=6505
DUSRL=6506
DUSRH=6507
GTF=6004
RTF=6005
RMF=6244
RIB=6234
SRQ=6003

FIXTAB
                *20

0020    0600    XMAG,       MAG
0021    0442    XWRITE,     WRITE
0022    0400    XREAD,      READ
0023    0456    XBUF,       BUF
0024    0000    SEQL,       0
0025    0000    SEQH,       0
0026    3003    FL,         3003
0027    3002    FH,         3002
0030    3004    MPF,        3004
0031    0000    CNT,        0
0032    0000    OTLOC,      0
0033    0000    TMP,        0
0034    0000    REOF,       0
0035    0000    OH,         0
0036    0000    OL,         0

0037    1177    C1177,      1177
0040    7703    CM75,       -75
0041    7376    CM402,      -402
0042    7304    CM474,      -474
0043    0041    C41,        41
0044    3777    C3777,      3777
0045    7700    C7700,      7700
0046    7772    CM6,        -6
0047    4001    C4001,      4001
0050    7776    CM2,        -2
```

```
                        *200
0200    7300    STO,    CLA CLL
0201    7404            OSR             /GET CHAR FOR SW
0202    0220            AND C77
0203    3221            DCA CHAR
0204    7000            NOP
0205    1237            TAD C1000
0206    1223    ST3,    TAD C20
0207    3624            DCA I L402      /SET READ COMMAND FOR UNIT
0210    4422            JMS I XREAD     /SEARCH FOR DESIRED CHAR IN REG LIN
0211    4226            JMS COMP
0212    5210            JMP .-2
0213    4421            JMS I XWRITE    /TRANSFER RELATIVE RECORDS ONLY
0214    4422            JMS I XREAD
0215    4226            JMS COMP
0216    5240            JMP STI
0217    5213            JMP .-4

0220    0077    C77,    77
0221    0000    CHAR,   0
0222    7000    C7000   7000
0223    0020    C20,    20
0224    0402    L402,   402
0225    3033    C3033,  3033

0226    0000    COMP,   0
0227    1625            TAD I C3033
0230    7000            NOP
0231    0220            AND C77
0232    7041            CIA
0233    1221            TAD CHAR
0234    7650            SNA CLA         /PROPER CHARACTER?
0235    2226            ISZ COMP        /YES
0236    5626            JMP I COMP      /NO

0237    1000    C1000,  1000

0240    1624    ST1,    TAD I L402
0241    0222            AND C7000
0242    1222            TAD C7000
0243    7640            SZA CLA         /UNIT 1 (REG LISTINGS)?
0244    5300            JMP ST2         /NO, CAPTIONS - DO BLANKS
0245    7332            7332            /YES, DO UNIT 2 NEXT (CAPTIONS)
0246    5206            JMP ST3

                        *277
0277    0000    CNTB,   0

0300    4423    ST2,    JMS I XBUF      /SET UP BLANK RECORD
0301    7300            CLA CLL
0302    1277            TAD CNTB
0303    7041            CIA
0304    3277            DCA CNTB        /SET UP BLANKS COUNTER
0305    4421            JMS I XWRITE    /OUTPUT BLANKS
0306    2277            ISZ CNTB
0307    5305            JMP .-2
0310    7402            HLT             /PROGRAM COMPLETE

                        *340
```

```
0340  7300    START,    CLA CLL
0341  7404              OSR              /GET # BLANKS FOR SW
0342  3277              DCA CNTR
0343  7402              HLT
0344  5200              JMP STO
                        *400
0400  0000    READ,     0
0401  4420              JMS I XMAG
0402  1020              1020      /COM
0403  3000              3000      /ADR
0404  7000              -1000     /WC
0405  0100              100       /EXT REG
0406  7402              HLT       /EOF
0407  5600              JMP I READ        /O
0410  4420              JMS I XMAG
0411  1010              1010
0412  0000              0
0413  0000              0
0414  0000              0
0415  7000              NOP
0416  7402              HLT
0417  7000              NOP
0420  0416    XNYT,     416
0421  4420    TEND,     JMS I XMAG
0422  0050              50
0423  0000              0
0424  0000              0
0425  0000              0
0426  7000              NOP
0427  4420              JMS I XMAG
0430  0010              10
0431  0000              0
0432  0000              0
0433  0000              0
0434  7000              NOP
0435  7402              HLT
0436  7300              CLA CLL
0437  3035              DCA OH
0440  3036              DCA OL
0441  7402              HLT
0442  0000    WRITE,    0
0443  4420              JMS I XMAG
0444  0040              40        /COM
0445  3000              3000      /ADR
0446  7000              -1000     /WC
0447  0100              100       /EXT REG
0450  7402              HLT       /EOF
0451  2036              ISZ OL
0452  5642              JMP I WRITE
0453  2035              ISZ OH
0454  5642              JMP I WRITE
0455  7402              HLT
```

```
0456  0000  BUF,      0
0457  7300            CLA CLL
0460  1303            TAD C2777
0461  3012            DCA 12
0462  1300            TAD CM1000
0463  3031            DCA CNT
0464  3412  NXB1      DCA I 12
0465  2031            ISZ CNT
0466  5264            JMP NXB1
0467  5656            JMP I BUF
0470  3012            DCA 12
0471  1301            TAD CM420
0472  3031            DCA CNT
0473  7240  NXB2      CLA CMA
0474  3412            DCA I 12
0475  2031            ISZ CNT
0476  5273            JMP NXB2
0477  5656            JMP I BUF

0500  7000  CM1000,   -1000
0501  7360  CM420,    -420
0502  3017  C3017,    3017
0503  2777  C2777,    2777
            PAGE
            /FORMAT:          JMS I XMAG
            /         COMMAND
            /         ADDRESS
            /         WORD COUNT
            /         EXTENSION REGISTER
            /         RETURN:  EOF
            /         RETURN:  NORMAL

0600  0000  MAG,      0
0601  1337            TAD MCM12
0602  3351            DCA REVCNT      /SET 10 RETRIES
0603  1600            TAD I MAG
0604  2200            ISZ MAG
0605  3352            DCA COM   /GET COMMAND
0606  7240            CLA CMA
0607  1600            TAD I MAG
0610  2200            ISZ MAG
0611  3353            DCA CADDR       /GET CURRENT ADDRESS
0612  1600            TAD I MAG
0613  2200            ISZ MAG
0614  3354            DCA WRDCNT      /GET WORD COUNT
0615  1600            TAD I MAG
0616  2200            ISZ MAG
0617  3355            DCA COMEX       /GET EXT REGISTER
0620  3350            DCA MADCOM
0621  1352  RETRY,    TAD COM
0622  1350            TAD MADCOM
0623  4322            JMS SETCOM      /SET CONTROLLER FUNCTION
0624  1354            TAD WRDCNT
0625  3032            DCA 32
0626  1353            TAD CADDR
0627  3033            DCA 33    /SET WORD COUNT & CURRENT ADDRESS
0630  1355            TAD COMEX
0631  6717            6717
0632  7300            CLA CLL   /SET EXT REG
0633  4332            JMS MAGOP       /PERFORM MAGTAPE FUNCTION
0634  6706            6706
0635  7421            MQL       /STORE STATUS IN MQ
0636  7501            MQA
```

```
0637  0340              AND MC6774
0640  7450              SNA
0641  5271              JMP MOK   /NO ERRORS
0642  0341              AND MC7677
0643  7450              SNA
0644  5273              JMP MEOF          /EOF
0645  0342              AND MC3543
0646  7640              SZA CLA
0647  7402              HLT       /BAD RECORD OFFLINE
0650  2351  PAR,        ISZ REVCNT        /PARITY
0651  5253              JMP .+2   /RETRY
0652  7402              HLT       /RETRY FAILURE
0653  1352              TAD COM
0654  0345              AND MC70
0655  1346              TAD MCM40
0656  7650              SNA CLA   /FAILURE ON MT?
0657  1343              TAD MC100         /YES
0660  3350              DCA MADCOM
0661  1352              TAD COM
0662  0344              AND MC7000
0663  1345              TAD MC70
0664  4322              JMS SETCOM        /SET CONTROLLER FOR BACKSPACE
0665  7240              CLA CMA
0666  3032              DCA 32    /WC = RECS BACKSPACES
0667  4332              JMS MAGOP         /PERFORM BACKSPACE
0670  5221              JMP RETRY         /AND TRY AGAIN
0671  2200  MOK,        ISZ MAG
0672  5600              JMP I MAG         /NORMAL EXIT
0673  1352  MEOF,       TAD COM
0674  0345              AND MC70
0675  1346              TAD MCM40
0676  7650              SNA CLA   /EOF ON WRITE?
0677  5250              JMP PAR   /YES - PARITY PROBLEM
0700  7000              NOP       /REWIND UNIT
0701  5600              JMP I MAG         /EXIT EOF
0702  0000  REWIND,     0
0703  1352              TAD COM
0704  0344              AND MC7000
0705  1347              TAD MC10
0706  4322              JMS SETCOM        /SET CONTROLLER FOR REWIND
0707  6722              6722      /EXECUTE REWIND
0710  1352              TAD COM
0711  0344              AND MC7000
0712  7106              CLL RTL
0713  7006              RTL       /LOAD CONT W/ NEXT MAG OP
0714  1357              TAD NEXM
0715  3356              DCA NEXCOM
0716  1756              TAD I NEXCOM
0717  6716              6716
0720  7300              CLA CLL
0721  5702              JMP I REWIND
0722  0000  SETCOM,     0
0723  6711              6711
0724  5323              JMP .-1
0725  6716              6716
0726  6721              6721
0727  5326              JMP .-1
0730  7300              CLA CLL
0731  5722              JMP I SETCOM
0732  0000  MAGOP,      0
0733  6722              6722
0734  6701              6701
0735  5334              JMP .-1
0736  5732              JMP I MAGOP
```

```
0737    7766    MCM12,      -12
0740    2764    MC6774,     2764
0741    7667    MC7667,     7667
0742    3543    MC3543,     3543
0743    0100    MC100,      100
0744    7000    MC7000,     7000
0745    0070    MC70,       70
0746    7740    MCM40,      -40
0747    0010    MC10,       10
0750    0000    MADCOM,     0
0751    0000    REVCNT,     0
0752    0000    COM,        0
0753    0000    CADDR,      0
0754    0000    WRDCNT,     0
0755    0000    COMEX,      0
0756    0000    NEXCOM,     0
0757    0760    NEXM,       NEXTM
0760    1020    NEXTM,      1020    /0
0761    0040                40      /1
0762    0040                40      /2
0763    0040                40      /3

BUF     0456
CADDR   0753
CHAR    0221
CM1000  0500
CM2     0050
CM402   0041
CM420   0501
CM474   0042
CM6     0046
CM75    0040
CNT     0031
CNTB    0277
COM     0752
COMEX   0755
COMP    0226
C1000   0237
C1177   0037
C20     0223
C2777   0503
C3017   0502
C3033   0225
C3777   0044
C4001   0047
C41     0043
C7000   0222
C77     0220
C7700   0045
FH      0027
FL      0026
LA02    0224
MADCOM  0750
MAG     0600
MAGOP   0732
MCM12   0737
MCM40   0746
MC10    0747
MC100   0743
MC3543  0742
MC6774  0740
```

```
MC70     0745
MC7000   0744
MC7677   0741
MEOF     0673
MOK      0671
MPF      0030
NEXCOM   0756
NEXM     0757
NEXTM    0760
NXB1     0464
NXB2     0743
OH       0035
OL       0036
OTLOC    0032
PAR      0650
READ     0400
REOF     0034
RETRY    0621
REVCNT   0751
REWIND   0702
SEQH     0025
SEQL     0024
SETCOM   0722
START    0340
ST0      0200
ST1      0240
ST2      0300
ST3      0206
TEND     0421
TMP      0033
WRDCNT   0754
WRITE    0442
XBUF     0023
XMAG     0020
XNYT     0420
XREAD    0022
XWRITE   0021
```

Program No. 2, Loader 2

```
BSW=7002
MQL=7421
MQA=7501
SWP=7521
CDF0=6201
CDF1=6211
CDF2=6221
DMCR=6517
DCBRH=6501
DCBRL=6500
DSSRH=6503
DSSRL=6502
DESR0=6511
DESR1=6515
DWCR=6516
DDMAR=6514
DCSR=6512
DAIRL=6504
DAIRH=6505
DUSRL=6506
```

```
                    DUSRH=6507
                    GTF=6004
                    RTF=6005
                    RMF=6244
                    RIB=6234
                    SRQ=6003

                    FIXTAB
                              *20

0020    0600    XMAG,       MAG
0021    0442    XWRITE,     WRITE
0022    0400    XREAD,      READ
0023    0456    XBUF,       BUF
0024    0000    SEQL,       0
0025    0000    SEQH,       0
0026    3003    FL,         3003
0027    3002    FH,         3002
0030    3004    MPF,        3004
0031    0000    CNT,        0
0032    0000    OTLOC,      0
0033    0000    TMP,        0
0034    0000    REOF,       0
0035    0000    OH,         0
0036    0000    OL,         0

0037    1177    C1177,      1177
0040    7703    CM75,       -75
0041    7376    CM402,      -402
0042    7304    CM474,      -474
0043    0041    C41,        41
0044    3777    C3777,      3777
0045    7700    C7700,      7700
0046    7772    CM6,        -6
0047    4001    C4001,      4001
0050    7776    CM2,        -2

                    PAGE

                              *200
0200    7300    ST0,        CLA CLL
0201    7404                OSR                 /GET CHAR FROM SW
0202    0220                AND C77
0203    3221                DCA CHAR
0204    7000                NOP
0205    1237                TAD C1000
0206    1223    ST3,        TAD C20
0207    3624                DCA I L402          /SET READ COMMAND FOR UNIT 1
0210    4422                JMS I XREAD         /SEARCH FOR DESIRED CHAR IN REG LIS
0211    4226                JMS COMP
0212    5210                JMP .-2
0213    4421                JMS I XWRITE        /TRANSFER RELATIVE RECORDS ONLY
0214    4422                JMS I XREAD
0215    4226                JMS COMP
0216    5240                JMP ST1
0217    5213                JMP .-4

0220    0077    C77,        77
0221    0000    CHAR,       0
0222    7000    C7000,      7000
0223    0020    C20,        20
0224    0402    L402,       402
0225    3033    C3033,      3033
```

```
0226  0000  COMP,     0
0227  1625            TAD I C3033
0230  7000            NOP
0231  0220            AND C77
0232  7041            CIA
0233  1221            TAD CHAR
0234  5320            JMP ST10
0235  2226            ISZ COMP          /YES
0236  5625            JMP I COMP        /NO

0237  1000  C1000,    1000

0240  1624  ST1,      TAD I L402
0241  0222            AND C7000
0242  1222            TAD C7000
0243  7640            SZA CLA           /UNIT 1 (REG LISTINGS)?
0244  5300            JMP ST2           /NO, CAPTIONS - DO BLANKS
0245  7332            7332              /YES, DO UNIT 2 NEXT (CAPTIONS)
0246  5206            JMP ST3

                      *277
0277  0000  CNTB,     0

0300  4423  ST2,      JMS I XBUF        /SET UP BLANK RECORD
0301  7300            CLA CLL
0302  1277            TAD CNTB
0303  7041            CIA
0304  3277            DCA CNTB          /SET UP BLANKS COUNTER
0305  4421            JMS I XWRITE      /OUTPUT BLANKS
0306  2277            ISZ CNTB
0307  5305            JMP .-2
0310  7402            HLT               /PROGRAM COMPLETE

                      *316

0316  3034  C3034,    3034
0317  0000  CHARX,    0

0320  7640  ST10,     SZA CLA           /MATCH?
0321  5626            JMP I COM         /NO
0322  1716            TAD I C3034
0323  7002            BSW               /CHECK 2ND CHAR
0324  0220            AND C77
0325  7041            CIA
0326  1317            TAD CHARX
0327  7640            SZA CLA           /2ND MATCH?
0330  2226            ISZ COMP          /NO, ACCEPT RECORD FOR TRANSFER
0331  5626            JMP I COMP        /YES, REJECT RECORD FOR TRANSFER

                      *340

0340  7300  START,    CLA CLL
0341  7404            OSR               /GET # BLANKS FOR SW
0342  3277            DCA CNTB
0343  7402            HLT
0344  5200            JMP ST0
                      *400
```

```
0400  0000   READ,     0
0401  4420             JMS I XMAG
0402  1020             1020      /COM
0403  3000             3000      /ADR
0404  7000             -1000     /WC
0405  0100             100       /EXT REG
0406  7402             HLT       /EOF
0407  5600             JMP I READ        /OH
0410  4420             JMS I XMAG
0411  1010             1010
0412  0000             0
0413  0000             0
0414  0000             0
0415  7000             NOP
0416  7402             HLT
0417  7000             NOP
0420  0416   XNYT,     416
0421  4420   TEND,     JMS I XMAG
0422  0050             50
0423  0000             0
0424  0000             0
0425  0000             0
0426  7000             NOP
0427  4420             JMS I XMAG
0430  0010             10
0431  0000             0
0432  0000             0
0433  0000             0
0434  7000             NOP
0435  7402             HLT
0436  7300             CLA CLL
0437  3035             DCA OH
0440  3036             DCA OL
0041  7402             HLT

0042  0000   WRITE,    0
0443  4420             JMS I XMAG
0444  0040             40        /COM
0445  3000             3000      /ADR
0446  7000             -1000     /WC
0447  0100             100       /EXT REG
0450  7402             HLT       /EOF
0451  2036             ISZ OL
0452  5642             JMP I WRITE
0453  2035             ISZ OH
0454  5642             JMP I WRITE
0455  7402             HLT

0456  0000   BUF,      0
0457  7300             CLA CLL
0460  1303             TAD C2777
0461  3012             DCA 12
0462  1300             TAD CM1000
0463  3031             DCA CNT
0464  3412   NXB1      DCA I 12
0465  2031             ISZ CNT
0466  5264             JMP NXB1
0467  5656             JMP I BUF
0470  3012             DCA 12
0471  1301             TAD CM420
0472  3031             DCA CNT
0473  7240   NXB2      CLA CMA
0474  3412             DCA I 12
0475  2031             ISZ CNT
0476  5273             JMP NXB2
0477  5656             JMP I BUF
```

```
0500  7000  CM1000,   -1000
0501  7660  CM420,    -420
0502  3017  C3017,    3017
0503  2777  C2777,    2777
            PAGE
            /FORMAT:            JMS I XMAG
            /         COMMAND
            /         ADDRESS
            /         WORD COUNT
            /         EXTENSION REGISTER
            /         RETURN:  EOF
            /         RETURN:  NORMAL

0600  0000  MAG,      0
0601  1337            TAD MCM12
0602  3351            DCA REVCNT        /SET 10 RETRIES
0603  1600            TAD I MAG
0604  2200            ISZ MAG
0605  3352            DCA COM  /GET COMMAND
0606  7240            CLA CMA
0607  1600            TAD I MAG
0610  2200            ISZ MAG
0611  3353            DCA CADDR         /GET CURRENT ADDRESS
0612  1600            TAD I MAG
0613  2200            ISZ MAG
0614  3354            DCA WORDCNT       /GET WORD COUNT
0615  1600            TAD I MAG
0616  2200            ISZ MAG
0617  3355            DCA COMEX         /GET EXT REGISTER
0620  3350            DCA MADCOM
0621  1352  RETRY,    TAD COM
0622  1350            TAD MADCOM
0623  4322            JMS SETCOM        /SET CONTROLLER FOR FUNCTION
0624  1354            TAD WRDCNT
0625  3032            DCA 32
0626  1353            TAD CADDR
0627  3033            DCA 33   /SET WORD COUNT & CURRENT ADDRESS
0630  1355            TAD COMEX
0631  6717            6717
0632  7300            CLA CLL  /SET EXT REG
0633  4332            JMS MAGOP         PERFORM MAGTAPE FUNCTION
0634  6706            6706
0635  7421            MQL      /STORE STATUS IN MQ
0636  7501            MQA
0637  0340            AND MC6774
0640  7450            SNA
0641  5271            JMP MOK  /NO ERRORS
0642  0341            AND MC7677
0643  7450            SNA
0644  5273            JMP MEOF          /EOF
0645  0342            AND MC3543
0646  7640            SZA CLA
0647  7402            HLT      /BAD RECORD OFFLINE
0650  2351  PAR,      ISZ REVCNT        /PARITY
0651  5253            JMP .+2  /RETRY
0652  7402            HLT      /RETRY FAILURE
0653  1352            TAD COM
0654  0345            AND MC70
0655  1346            TAD MCM40
0656  7650            SNA CLA  /FAILURE ON MT?
0657  1343            TAD MC100         /YES
```

```
0660  3350              DCA MADCOM
0661  1352              TAD COM
0662  0344              AND MC7000
0663  1345              TAD MC70
0664  4322              JMS SETCOM        /SET CONTROLLER FOR BACKSPACE
0665  7240              CLA CMA
0666  3032              DCA 32   /WC = RECS BACKSPACED
0667  4332              JMS MAGOP         /PERFORM BACKSPACE
0670  5221              JMP RETRY         /AND TRY AGAIN
0671  2200  MOK,        ISZ MAG
0672  5600              JMP I MAG         /NORMAL EXIT
0673  1352  MEOF,       TAD COM
0674  0345              AND MC70
0675  1346              TAD MCM40
0676  7650              SNA CLA  /EOF ON WRITE?
0677  5250              JMP PAR  /YES - PARITY PROBLEM
0700  7000              NOP      /REWIND UNIT
0701  5600              JMP I MAG         /EXIT EOF
0702  0000  REWIND,     0
0703  1352              TAD COM
0704  0344              AND MC7000
0705  1347              TAD MC10
0706  4322              JMS SETCOM        /SET CONTROLLER FOR REWIND
0707  6722              6722     /EXECUTE REWIND
0710  1352              TAD COM
0711  0344              AND MC7000
0712  7106              CLL RTL
0713  7006              RTL      /LOAD CONT W/ NEXT MAG OP
0714  1357              TAD NEXM
0715  3356              DCA NEXCOM
0716  1756              TAD I NEXCOM
0717  6716              6716
0720  7300              CLA CLL
0721  5702              JMP I REWIND
0722  0000  SETCOM,     0
0723  6711              6711
0724  5323              JMP .-1
0725  6716              6716
0726  6721              6721
0727  5326              JMP .-1
0730  7300              CLA CLL
0731  5722              JMP I SETCOM
0732  0000  MAGOP,      0
0733  6722              6722
0734  6701              6701
0735  5334              JMP .-1
0736  5732              JMP I MAGOP
0737  7766  MCM12,      -12
0740  2764  MC6774,     2764
0741  7667  MC7677,     7677
0742  3543  MC3543,     3543
0743  0100  MC100,      100
0744  7000  MC7000,     7000
0745  0070  MC70,       70
0746  7740  MCM40,      -40
0747  0010  MC10,       10
0750  0000  MADCOM,     0
0751  0000  REVCNT,     0
0752  0000  COM,        0
0753  0000  CADDR,      0
0754  0000  WRDCNT,     0
0755  0000  COMEX,      0
0756  0000  NEXCOM,     0
0757  0760  NEXM,       NEXTM
0760  1020  NEXTEM,     1020      /0
0761  0040              40        /1
0762  0040              40        /2
0763  0040              40        /3
```

CADDR	0753
CHAR	0221
CHARX	0317
CM1000	0500
CM2	0050
CM402	0041
CM420	0501
CM474	0042
CM6	0045
CM75	0040
CNT	0031
CNTB	0277
COM	0752
COMEX	0755
COMP	0226
C1000	0237
C1177	0037
C20	0223
C2777	0503
C3017	0502
C3033	0225
C3034	0316
C3777	0044
C4001	0047
C41	0043
C7000	0222
C77	0220
C7700	0045
FH	0027
FL	0026
L402	0224
MADCOM	0750
MAG	0600
MAGOP	0732
MCM12	0737
MCM40	0746
MC10	0747
MC100	0743
MC3543	0742
MC6774	0740
MC70	0745
MC7000	0744
MC7677	0741
MEOF	0673
MOK	0671
MPF	0030
NEXCOM	0756
NEXM	0757
NEXTM	0760
NXB1	0464
NXB2	0473
OH	0035
OL	0036
OTLOC	0032
PAR	0650
READ	0400
REOF	0034
RETRY	0621
REVCNT	0751
REWIND	0702
SEQH	0025

```
SEQL    0024
SETCOM  0722
START   0340
ST0     0200
ST1     0240
ST10    0320
ST2     0300
ST3     0206
TEND    0421
TMP     0033
WRDCNT  0754
WRITE   0442
XBUF    0023
XMAG    0020
XNXT    0420
XREAD   0022
XWRITE  0021
```


Program No. 2,. Loader 3

```
        BSW=7002
        MQL=7421
        MQA=7501
        SWP=7521
        CDF0=6201
        CDF1=6211
        CDF2=6221
        DMCR=6517
        DCBRH=6501
        DCBRL=6500
        DSSRH=6503
        DSSRL=6502
        DESR0=6511
        DESR1=6515
        DWCR=6516
        DDMAR=6514
        DCSR=6512
        DAIRL=6504
        DAIRH=6505
        DUSRL=6506
        DUSRH=6507
        GTF=6004
        RTF=6005
        RMF=6244
        RIB=6234
        SRQ=6003

        FIXTAB
                *20

0020  0600  XMAG,   MAG
0021  0442  XWRITE, WRITE
0022  0400  XREAD,  READ
0023  0456  XBUF,   BUF
0024  0000  SEQL,   0
0025  0000  SEQH,   0
0026  3003  FL,     3003
0027  3002  FH,     3002
0030  3004  MPF,    3004
0031  0000  CNT,    0
0032  0000  OTLOC,  0
0033  0000  TMP,    0
0034  0000  REOF,   0
0035  0000  OH,     0
0036  0000  OL,     0
```

```
0037  1177  C1177,  1177
0040  7703  CM75,   -75
0041  7376  CM402,  -402
0042  7304  CM474,  -474
0043  0041  C41,    41
0044  3777  C3777,  3777
0045  7700  C7700,  7700
0046  7772  CM6,    -6
0047  4001  C4001,  4001
0050  7776  CM2,    -2


            PAGE

                    *200
0200  7300  ST0,    CLA CLL
0201  7404          OSR             /GET CHAR FROM SW
0202  0220          AND C77
0203  3221          DCA CHAR
0204  7000          NOP
0205  1237          TAD C1000
0206  1223  ST3,    TAD C20
0207  3624          DCA I L402      /SET READ COMMAND FOR UNIT 1
0210  4422          JMS I XREAD     /SEARCH FOR DESIRED CHAR IN REG. LI
0211  4226          JMS COMP
0212  5210          JMP .-2
0213  4421          JMS I XWRITE    /TRANSFER RELATIVE RECORDS ONLY
0214  4422          JMS I XREAD
0215  4226          JMS COMP
0216  5240          JMP ST1
0217  5213          JMP .-4

0220  0077  C77,    77
0221  0000  CHAR,   0
0222  7000  C7000,  7000
0223  0020  C20,    20
0224  0402  L402,   402
0225  3033  C3033,  3033

0226  0000  COMP,   0
0227  1625          TAD I C3033
0230  7000          NOP
0231  0220          AND C77
0232  7041          CIA
0233  1221          TAD CHAR
0234  5320          JMP ST10
0235  2226          ISZ COMP        /YES
0236  5626          JMP I COMP      /NO

0237  1000  C1000,  1000

0240  1624  ST1,    TAD I L402
0241  0222          AND C7000
0242  1222          TAD C7000
0243  7640          SZA CLA         /UNIT 1 (REG LISTINGS)?
0244  5300          JMP ST2         /NO, CAPTIONS - DO BLANKS
0245  1251          TAD C5320       /SET UP 2ND CHAR COMPARE
0246  3234          DCA 234
0247  7332          7332
0250  5206          JMP 206
0251  5320  C5320,  5320
```

```
                        *277
0277  0000  CNTB,       0

0300  4423  ST2,        JMS I XBUF          /SET UP BLANK RECORD
0301  7300               CLA CLL
0302  1277               TAD CNTB
0303  7041               CIA
0304  3277               DCA CNTB            /SET UP BLANKS COUNTER
0305  4421               JMS I XWRITE        /OUTPUT BLANKS
0306  2277               ISZ CNTB
0307  5305               JMP .-2
0310  7402               HLT                 /PROGRAM COMPLETE

                        *316

0316  3034  C3034,      3034
0317  0000  CHARX,      0

0320  7640  ST10,       SZA CLA             /MATCH?
0321  5626               JMP I COMP          /NO
0322  1716               TAD I C3034         /2ND CHAR CHECK
0323  7002               BSW
0324  0220               AND C77
0325  7041               CIA
0326  1317               TAD CHARX
0327  7640               SZA CLA             /2ND MATCH?
0330  5626               JMP I COMP          /NO, REJECT RECORD
0331  1335               TAD 335
0332  3234               DCA 234
0333  2226               ISZ COMP            /RESET NORMATL EXIT TO COMPARE SUB
0334  5626               JMP I COMP
0335  7650               7650

                        *340

0340  7300  START,      CLA CLL
0341  7404               OSR                 /GET # BLANKS FROM SW
0342  3277               DCA CNTB
0343  7402               HLT
0344  5200               JMP ST0
                        *400
0400  0000  READ,       0
0401  4420               JMS I XMAG
0402  1020               1020     /COM
0403  3000               3000     /ADR
0404  7000               -1000    /WC
0405  0100               100      /EXT REG
0406  7402               HLT      /EOF
0407  5600               JMP I READ          /OK
0410  4420               JMS I XMAG
0411  1010               1010
0412  0000               0
0413  0000               0
0414  0000               0
0415  7000               NOP
0416  7402               HLT
0417  7000               NOP
0420  0416  XNYT,       416
```

```
0421  4420  TEND,   JMS I XMAG
0422  0050          50
0423  0000          0
0424  0000          0
0425  0000          0
0426  7000          NOP
0427  4420          JMS I XMAG
0430  0010          10
0431  0000          0
0432  0000          0
0433  0000          0
0434  7000          NOP
0435  7402          HLT
0436  7300          CLA CLL
0437  3035          DCA OH
0440  3036          DCA OL
0441  7402          HLT.

0442  0000  WRITE,  0
0443  4420          JMS I XMAG
0444  0040          40       /COM
0445  3000          3000     /ADR
0446  7000          -1000    /WC
0447  0100          100      /EXT REG
0450  7402          HLT      /EOF
0451  2036          ISZ OL
0452  5642          JMP I WRITE
0453  2035          ISZ OH
0454  5642          JMP I WRITE
0455  7402          HLT

0456  0000  BUF,    0
0457  7300          CLA CLL
0460  1303          TAD C2777
0461  3012          DCA 12
0462  1300          TAD CM1000
0463  3031          DCA CNT
0464  3412  NXB1,   DCA I 12
0465  2031          ISZ CNT
0466  5264          JMP NXB1
0467  5656          JMP I BUF
0470  3012          DCA 12
0471  1301          TAD CM420
0472  3031          DCA CNT
0473  7240  NXB2,   CLA CMA
0474  3412          DCA I 12
0475  2031          ISZ CNT
0476  5273          JMP NXB2
0477  5656          JMP I BUF

0500  7000  CM1000, -1000
0501  7360  CM420,  -420
0502  3017  C3017,  3017
0503  2777  C2777,  2777     /
              PAGE
              /FORMAT:        JMS I XMAG
              /       COMMAND
              /       ADDRESS
              /       WORD COUNT
              /       EXTENSION REGISTER
              /       RETURN: EOF
              /       RETURN: NORMAL
```

```
0600  0000  MAG,    0
0601  1337          TAD MCM12
0602  3351          DCA REVCNT      /SET 10 RETRIES
0603  1600          TAD I MAG
0604  2200          ISZ MAG
0605  3352          DCA COM /GET COMMAND
0606  7240          CLA CMA
0607  1600          TAD I MAG
0610  2200          ISZ MAG
0611  3353          DCA CADDR       /GET CURRENT ADDRESS
0612  1600          TAD I MAG
0613  2200          ISZ MAG
0614  3354          DCA WRDCNT      /GET WORD COUNT
0615  1600          TAD I MAG
0616  2200          ISZ MAG
0617  3355          DCA COMEX       /GET EXT REGISTER
0620  3350          DCA MADCOM
0621  1352  RETRY,  TAD COM
0622  1350          TAD MADCOM
0623  4322          JMS SETCOM      /SET CONTROLLER FOR FUNCTION
0624  1354          TAD WRDCNT
0625  3032          DCA 32
0626  1353          TAD CADDR
0627  3033          DCA 33  /SET WORD COUNT & CURRENT ADDRESS
0630  1355          TAD COMEX
0631  6717          6717
0632  7300          CLA CLL /SET EXT REG
0633  4332          JMS MAGOP       /PERFORM MAGTAPE FUNTION
0634  6706          6706
0635  7421          MQL     /STORE STATUS IN MQ
0636  7501          MQA
0637  0340          AND MC6774
0640  7450          SNA
0641  5271          JMP MOK /NO ERRORS
0642  0341          AND MC7677
0643  7450          SNA
0644  5273          JMP MEOF        /EOF
0645  0342          AND MC3543
0646  7640          SZA CLA
0647  7402          HLT     /BAD REC OR OFFLINE
0650  2351  PAR,    ISZ REVCNT      /PARITY
0651  5253          JMP .+2 /RETRY
0652  7402          HLT     /RETRY FAILURE
0653  1352          TAD COM
0654  0345          AND MC70
0655  1346          TAD MCM40
0656  7650          SNA CLA /FAILURE ON MT?
0657  1343          TAD MC100       /YES
0660  3350          DCA MADCOM
0661  1352          TAD COM
0662  0344          AND MC7000
0663  1345          TAD MC70
0664  4322          JMS SETCOM      /SET CONTROLLER FOR BACKSPACE
0665  7240          CLA CMA
0666  3032          DCA 32  /WC = RECS BACKSPACED
0667  4332          JMS MAGOP       /PERFORM BACKSPACE
0670  5221          JMP RETRY       /AND TRY AGAIN
0671  2200  MOK,    ISZ MAG
0672  5600          JMP I MAG       /NORMAL EXIT
0673  1352  MEOF,   TAD COM
0674  0345          AND MC70
0675  1346          TAD MCM40
0676  7650          SNA CLA /EOF ON WRITE?
0677  5250          JMP PAR /YES - PARITY PROBLEM
0700  7000          NOP     /REWIND UNIT
0701  5600          JMP I MAG       /EXIT EOF
```

```
0702  0000  REWIND, 0
0703  1352         TAD COM
0704  0344         AND MC7000
0705  1347         TAD MC10
0706  4322         JMS SETCOM       /SET CONTROLLER FOR REWIND
0707  6722         6722     /EXECUTE REWIND
0710  1352         TAD COM
0711  0344         AND MC7000
0712  7106         CLL RTL
0713  7006         RTL      /LOAD CONT W/ NEXT MAG OP
0714  1357         TAD NEXM
0715  3356         DCA NEXCOM
0716  1756         TAD I NEXCOM
0717  6716         6716
0720  7300         CLA CLL
0721  5702         JMP I REWIND
0722  0000  SETCOM, 0
0723  6711         6711
0724  5323         JMP .-1
0725  6716         6716
0726  6721         6721
0727  5326         JMP .-1
0730  7300         CLA CLL
0731  5722         JMP I SETCOM
0732  0000  MAGOP,  0
0733  6722         6722
0734  6701         6701
0735  5334         JMP .-1
0736  5732         JMP I MAGOP
0737  7766  MCM12,  -12
0740  2764  MC6774, 2764
0741  7667  MC7677, 7667
0742  3543  MC3543, 3543
0743  0100  MC100,  100
0744  7000  MC7000, 7000
0745  0070  MC70,   70
0746  7740  MCM40,  -40
0747  0010  MC10,   10
0750  0000  MADCOM, 0
0751  0000  REVCNT, 0
0752  0000  COM,    0
0753  0000  CADDR,  0
0754  0000  WRDCNT, 0
0755  0000  COMEX,  0
0756  0000  NEXCOM, 0
0757  0760  NEXM,   NEXTM
0760  1020  NEXTM,  1020    /0
0761  0040         40       /1
0762  0040         40       /2
0763  0040         40       /3


BUF    0456
CADDR  0753
CHAR   0221
CHARX  0317
CM1000 0500
CM2    0050
CM402  0041
CM420  0501
CM474  0042
CM6    0046
CM75   0040
CNT    0031
CNTB   0277
COM    0752
COMEX  0755
```

```
COMP    0226
C1000   0237
C1177   0037
C20     0223
C2777   0503
C3017   0502
C3033   0225
C3034   0316
C3777   0044
C4001   0047
C41     0043
C5320   0251
C7000   0222
C77     0220
C7700   0045
FH      0027
FL      0026
L402    0224
MADCOM  0750
MAG     0600
MAGOP   0732
MCM12   0737
MCM40   0746
MC10    0747
MC100   0743
MC3543  0742
MC6774  0740
MC70    0745
MC7000  0744
MC7677  0741
MEOF    0673
MOK     0671
MPF     0030
NEXCOM  0756
NEXM    0757
NEXTM   0760
NXB1    0464
NXB2    0473
OH      0035
OL      0036
OTLOC   0032
PAR     0650
READ    0400
REOF    0034
RETRY   0621
REVCNT  0751
REWIND  0702
SEOH    0025
SEOL    0024
SETCOM  0722
START   0340
ST0     0200
ST1     0240
ST10    0320
ST2     0300
ST3     0206
TEND    0421
TMP     0033
WRDCNT  0754
WRITE   0442
XBUF    0023
XMAG    0020
XNYT    0420
XREAD   0022
XWRITE  0021
.F
```

Operating Instructions for the above listed Program No. 2 are as follows:

1. Set up and load magnetic tapes. Load program into memory bank 0. Add toggles for leaders 2 & 3 loc. 317=2nd char. which defines split.

   Listings-tape unit 1

   Captions-tape unit 2

   Scratch-tape unit 0

2. Set Switch Register (SR)=340: load and clear
3. Set SR=# of blank records desired: continue. When program halts (almost instantly) set SR=DRC letter being processed (bits 6-11): continue. The program will halt at location 310 after it has completed the appropriate merge function of the desired alpha grouping:

   a.) To terminate unit- Set SR=421: load and continue. Program halts at location 435 (program complete).

   b.) To terminate output tape only- Set SR=421: load and continue. Program halts at location 435. Go to instruction #2 to continue processing.

   c.) To continue on same output tape- Go to instruction #2 to continue processing.

Halts: 310- see above

406- end of file on input tape. Manually rewind and dismount tape; mount and load next tape on tape drive; SR=401; load, continue.

450, 647, 652- Magtape failures

C. Retrieval File and Base Data File Construction

C1. In General.

This portion of the exemplary embodiment is generally depicted in FIGURE 25. It comprises Program numbers 3, 4 and 5. Program 5 serves to create an ASCII coded base data magnetic disk file from the standardized converted output 158 of the conversion portion of the system. Program 3 takes this same input data and constructs an intermediate retrieval file on magnetic tape while program 4 uses the intermediate file to temporarily construct a miniature version of a retrieval file before finally constructing the magnetic disk retrieval file. An ASCII base data file and a corresponding retrieval file are constructed and stored on magnetic disk for each of the 16 alphabetic groupings or setments previously discussed.

For this exemplary embodiment, the following set of PICs is used:

Field 1. In Finding Name Field (first full word of standardized name field) and;

$\phi_{2,2+}$; $\phi_{3,3+}$; $\phi_{4,4+}$; $\phi_{5,5+}$; $\phi_{6,6+}$;

$\phi_{7,7+}$ where ϕ = A → Z (regardless of case)

[Note: The first character value is ignored because the files are already separated based on the first character of this field.]

Field 2. In name field but not finding name (all other words of name field plus full title field) and;

$\phi_{1,1+}$; $\phi_{2,2+}$; $\phi_{3,3+}$; $\phi_{4,4+}$; $\phi_{5,5+}$;

$\phi_{6,6+}$ where ϕ = A → Z (regardless of case)

Field 3. In designation field or street, house number house number suffix, locality fields and

$\phi_{1,1+}$; $\phi_{2,2+}$; $\phi_{3,3+}$; $\phi_{4,4+}$ where

ϕ = A→ Z (regardless of case) and ϕ = 0→ 9

Field 4. In business-professional listings or in residential-professional listings.

Thus there are nominally 457 PICs in all associated with the alphabetic values and sequential locations of characters in each base data file record in this exemplary embodiment.

Program No. 3 first analyzes the above three noted fields and sets up a two-dimensional intermediate retrieval file comprising a bit matrix representing character value versus character position for each field of a given record. The binary bit values in this intermediate file are then transferred by Program 4 into a properly organized retrieval file (one array per PIC, one array element per base data file record, etc., before being transferred to magnetic disk for actual use. The format of the intermediated retrieval file matrix is shown in FIGURE 26.

Here in FIGURE 26, the bit matrix is shown as comprising several successive 12 bit words in magnetic core storage. Each of the 457 significant bits is indicated by a decimal numbered grid opening and is equivalent directly to the octal number (after the usual decimal-octal conversion) of a respectively corresponding array in the retrieval file.

The organization of the 16 data base files and associated retrieval files is depicted in FIGURES 27 and 28 for the two disk drives involved in this exemplary embodiment.

As should now be apparent, each alpha segment has 500 disk blocks reserved for storage of the ASCII-coded base data file records. Each block contains 60 fixed length records. Each record being 40 words in length (80 six bit characters). Note that the blocks and listings within the block are written in reverse order so that first in will be last out. Also, move cursor commands are embedded in the data before the telephone number for better handling in the retrieval program (program #6). A standardized six bit coding is used except for control codes. A control code is a double coded character- 0 followed by a six bit code which if 200 is added to it will produce a CRT control character.

Standardized fields transferred to the magnetic disk base data file are:

Listing Name
Listing Title
Listing Designation
House #
House # Suffix
Street
Locality
Telephone #, NPA, Non Pub Information

A brief functional description of program numbers 5, 3 and 4 follows as an introduction to a more detailed description of each:

1. Program #5

A. Obtains the necessary fields for CRT display at time of retrieval.

1. Listing name.
2. Listing title.
3. Listing designation.
4. Street
5. House Number
6. House Number suffix
7. Locality
8. NPA
9. Telephone number; non-pub info

B. Format to retrieval specifications.

1. Fixed length records, blocked, stripped ASCII coding (6 bit) packed
2. Diva Disk output; fixed length records and blocks; 60 records per block; 40 computer words per record; maximum of 80 characters per record. The data blocks occupy 500 disk blocks max. for each alpha group. Thus allowing 30,000 listings per alpha segment.

2. Program #3

A. Obtains the necessary fields for XM searching capability.

1. Full name.
2. Full title.
3. Full designation.
4. Street.
5. House number.
6. House number suffix.

7. Locality.
8. Listing Type.

B. Format to intermediate retrieval file specifications

1. The data is entered into a bit matrix, the matrix being a character versus character position relocation.
2. The data is separated into 4 fields.
   a. Finding name- first word of full name field
   b. Subsequent words- other words of full name field plus full title field.
   c. address- full designation, street, house number, house number suffix and locality fields.
   d. Listing Type
      (i) business-professional
      (ii) residential-professional
3. Output to magtape intermediate retrieval file blocks, 12 records per block, 64 computer words per record, 9 track, 1600 bpi, special core dump mode.

3. Program #4

This program takes 12 intermediate retrieval file records and makes one retrieval file word for each character/character position element. These are written onto the small disk (DEC DF32D) until 120 frames (10 computer words) are stored for each element. The disk is then dumped onto

magnetic tape (intermediate process). After all intermediate retrieval files are processed in this way, these mini retrieval files (120 bits each) are then input from tape and written at the proper block on the Diva Disk. Note that the intermediate file bits are processed in reverse order so that last in will be first out at retrieval time. The retrieval files occupy 456 disk blocks per alpha group; each retrieval file thus allows thirty thousand bit positions.

C2. Detailed Description of Program No. 5

Program No. 5 is shown in block form at FIGURES 29-32 with subroutines utilized therein detailed in FIGURES 33-43. An explicit listing of the assembly level source program language for Program No. 5 and all related subroutines follows. It will be noted that the main program is shown as comprising sections I, II and III in FIGURE 29, IV in FIGURE 30, V in FIGURE 31 and VI in FIGURE 32. These program sections correspond to specific listing statements:

Section	Instruction Statement
I	0400-0412
II	0413-0420; 0554-0560
III	0421-0434
IV	0435-0446; 0526-0536; 0561-0564
V	0447-0465; 0515-0525; 0565-0573
VI	0466-0514

These six program sections are functionally described below as an introduction to the explicit program listing:

I. Initialize.

A. Set up the starting disk output block (which is manually entered before program initiation). Note that the blocks are processed in reversed order (as are each record within a block) so that last in will be first out.

B. XSTDSK: set controller and drive #6.

C. XZRO: zero entire output block.

D. XINIT: Set up starting output record pointer (see above); set up the buffer which will store the phone #, NPA, non-pub info, and cursor commands until the end of the record is ready for processing; set up output record per block counter (60 records per block).

E. Indicate that the 1st character will be put in the 1st half of the output buffer word. Reset the character counter for the line. Set the space indicator (for deciding whether the current character is the 1st character of a word).

II. Input.

A. Read record from magnetic tape.

B. Halt on eof, dump last output block manually and halt.

III. Processing.

A. Set output data pointer at beginning of next record.

B. If blank record- ignore any data processing.

C. Check for full output block and output block to disk if full. Then go to process next record (rezeroing buffer if output transfer has occurred).

IV. Processing.

A. XSTORIT: set up output character "linct". Store telephone number characters- npa and either NP (for non pub) or telephone number digits.

B. Set up the list of data fields to be processed. Inclusive in this list are the following fields: listing name, listing title, listing designation, house #, house # suffix, street and locality. Process these fields into the output record then transfer the stored telephone information to the output record.

V. Processing.

Process a field. All spaces between words are removed within a field (with the exception of full 1) and each field is separated by a space except the telephone field which is positioned by move cursor characters.

IV. Processing.

Process double coded characters. Upon encountering a "start of field code", put a space into output record and go to V to process the next field. Also, make sure that the input area is not being overrun. For regular double coded characters, transfer then directly to the output record.

Some of the subroutines not shown in the drawings are briefly explained below:

DSK Subroutine

A general purpose subroutine which drives the disk hardware to write a block (2518_{10} words) of data onto the disk at a specified block. (The Disk contains possible 4060 blocks per sector and has 2 sectors for this program).

format: JMS I XDSK
(block #)
control resumes here

DIVIDE Subroutine

A general purpose single precision divide subroutine, used by "dsk" to determine the proper head and track from the block # (block # 20_{10})

format: JMS I XDIVID
(divident)
(divisor)
remainder returned here
control resumes here with quotient in accumulator.

MAG Subroutine

A general purpose subroutine to drive the magnetic tape transport.

format: JMS I XMAG *1
(command)
(current address)
(word count)
(extension register) *2
control resumes here if tape mark encountered
control resumes here normally.

---

*1: command bits	*2: extension bits
0- formatter select	0-4- N/A
1,2- unit select	5- special core dump mode
3-5- N/A	6-9- N/A
6-8- tape command	10-11- memory bank
9-11- N/A	

OCDEC Subroutine

A general purpose octal to decimal conversion subroutine (double precision).

```
format:     JMS I XOCDEC
5           (high order octal #)
            (low order octal #)
            ten millions digit
            million's digit
            hundred thousand's digit        equivalent digits
10          ten thousand's digit            returned here
            thousand's digit
            humdred's digit
            ten's digit
            unit's digit
15          control resumes here
```

Program No. 5

```
            CDF2=6221
            CDF1=6211
            CDF0=6201
            BSW=7002
            MQL=7421
            MQA=7501
            MCR=6517
            CBRH=6501
            CBRL=6500
            SSRH=6503
            SSRL=6502
            ESR0=6511
            WCR=6516
            MAR=6514
            CSR=6512
            AIRH=6505
            AIRL=6504
            FIXTAB

                        *20     /RETRY FAILURE RECOVERY
0020    1063            TAD STOUT
0021    1102            TAD CM50
0022    3063            DCA STOUT
0023    5424            JMP I NNN
0024    0431    NNN,    N
```

```
                        *40       /ADDRESS TABLE
0040    7021    C3020,  7021
0041    7655    CM123,  -123
0042    4020    C4020,  4020
0043    0077    C77,    77
0044    0002    L2,     2
0045    0003    L3,     3


0046    1550    XOTPUT, OTPUT     /TO DISK
0047    0200    XMAG,   MAG
0050    1400    XZRO,   ZRO
0051    1430    XINIT,  INIT
0052    5000    XDIVID, DIVIDE
0053    2600    XGETCR, GETCR
0054    1600    XPUTWD, PUTWD     /PUT STORED WD IN OUTBUF
0055    0302    XREWD,  REWIND
0056    1610    XPUTCR, PUTCR


0057    0112    XSTORE, WORD
0060    0000    STORE,  0
0061    0137    END,    WORD+25
0062    0000    OUTBUF, 0
0063    0000    STOUT,  0         /SET LOC FOR 1 DATA
0064    0000    LINE,   0         /1 IF LINE 2 OF DATA
0065    0000    HALF,   0         /PART OF OUTBUF WD
0066    0000    DUMP,   0
0067    0000    RECCT,  0
0070    0000    CHAR,   0
0071    0000    DBLK,   0
0072    0000    SEC,    0
0073    0000    DRIVE,  0
0074    0000    LINCT,  0

0075    3020    L3020,  3020
0076    4000    C4000,  4000
0077    7706    CM72,   -72
0100    7701    CM77,   -77
0101    7772    CM6,    -6
0102    7730    CM50,   -50
0103    7746    CM32,   -32
0104    7713    CM65,   -65
0105    0000    FLDTRG, 0
0106    0000    SPTRG,  0
0107    0000    UPTRG,  0
0110    1700    XSTDSK, SETDSK
0111    1000    XSTRT,  STORIT

0112    0000    WORD,   0         /ST OF WORD IN STORAGE- 60 CHAR LIMIT
                        *1400
1400    0000    ZRO,    0         /ZERO OUTBUF; 0-4725 MB2
1401    7300            CLA CLL
1402    1214            TAD CM4726        /-2518
1403    3215            DCA CTR
1404    3062            DCA OUTBUF
1405    6221            CDF2
1406    3462    AG1,    DCA I OUTBUF
1407    2062            ISZ OUTBUF
1410    2215            ISZ CTR
1411    5206            JMP AG1
1412    6201            CDF0
1413    5600            JMP I ZRO         /YES
1414    3052    CM4726, -4726
1415    0000    CTR,    0
```

```
                        *1430
1430    0000    INIT,   0
1431    1240            TAD C4726       /-2518
1432    3063            DCA STOUT       /END OF OUTBUF 1
1433    1057            TAD XSTORE
1434    3060            DCA STORE       /ST OF WORD STORAGE
1435    1241            TAD CM74        /-60
1436    3067            DCA RECCT       /60 REC PER BLOCK
1437    5630            JMP I INIT
1440    4726    C4726,  4726
1441    7704    CM74,   -74
                        *1550
1550    0000    OTPUT,  0
1551    7300            CLA CLL  /OUTPUT TO DISK FROM MB2
1552    1071            TAD DBLK
1553    3355            DCA DDBLK
1554    4767            JMS I XDSK
1555    0000    DDBLK,  0       /DISK BLOCK NO 40-1039
1556    7240            CLA CMA
1557    1071            TAD DBLK
1560    3071            DCA DBLK        /SET BACK ONE BLOCK
1561    1071            TAD DBLK
1562    1366            TAD CM4763      /-3059 - FIRST LIMIT
1563    7710            SPA CLA  /OVERFLOW?
1564    7402            HLT     /YES
1565    5750            JMP I OTPUT     /NO
1566    2015    CM5763, -5763   /SET EVERY TIME TO BLOCK LIMIT
1567    2000    XDSK,   DSK
                        *1600
1600    0000    PUTWD,  0       /PUT WORD IN OUTBUF
1601    7300            CLA CLL
1602    1460    AG2,    TAD I STORE     /GET CHAR
1603    7450            SNA     /DONE?
1604    5244            JMP QQ          /YES
1605    4210            JMS PUTCR       /NO, PUT CHAR INTO BUF
1606    2060            ISZ STORE
1607    5202            JMP AG2


1610    0000    PUTCR,  0
1611    3243            DCA CHR  /SAVE 6 BIT CHAR
1612    6221            CDF2
1613    1065            TAD HALF
1614    7640            SZA CLA  /CHAR IN 1ST HALF OF WD?
1615    5234            JMP SECHLF      /NO
1616    2065            ISZ HALF        /YES, SET TRG
1617    1243            TAD CHR
1620    7002            BSW
1621    3462            DCA I OUTBUF    /PUT IN BUF
1622    1243            TAD CHR
1623    1100            TAD CM77
1624    7650            SNA CLA         /SPACE?
1625    7001            IAC             /YES
1626    3107            DCA UPTRG       /NO
1627    6201    TG,     CDF0
1630    2074            ISZ LINCT       /BUF FULL?
1631    5610            JMP I PUTCR     /NO
1632    2210            ISZ PUTCR       /YES
1633    5610            JMP I PUTCR
1634    1243    SECHLF, TAD CHR
1635    1462            TAD I OUTBUF
1636    3462            DCA I OUTBUF    /PUT CHAR INTO SECOND HALF
1637    3065            DCA HALF        /CLEAR TRG
1640    2062            ISZ OUTBUF
1641    5222            JMP TG-5
1642    7402            HLT
1643    0000    CHR,    0
1644    1057    QQ,     TAD XSTORE
1645    3060            DCA STORE
1646    5600            JMP I PUTWD
```

```
                        *2000
2000    0000    DSK,    0
2001    7300            CLA CLL
2002    1600            TAD I DSK
2003    2200            ISZ DSK
2004    3206            DCA DIVR
2005    4452            JMS I XDIVID
2006    0000    DIVR,   0       /BLOCK NO
2007    0024            24      /DIVISOR - 20
2010    0000    HD,     0       /REMAINDER
2011    3333            DCA CYL  /CYL IN AC
2012    7326            CLA STL RTL     /2
2013    6517            MCR
2014    7300            CLA CLL  /LOAD MODE
2015    1330            TAD C1000
2016    6501            CBRH     /SELECT CYLINDER
2017    7300            CLA CLL
2020    1333            TAD CYL
2021    7106            CLL RTL
2022    7006            RTL
2023    6500            CBRL     /CYL NO IN BITS 0-7
2024    7300            CLA CLL
2025    1331            TAD C1400
2026    6501            CBRH     /SP FN
2027    7300            CLA CLL
2030    1331            TAD C1400
2031    6500            CBRL     /SEEK AND RESET HEAD
2032    7300            CLA CLL
2033    1210            TAD HD   /SET UP HEADER IMAGE
2034    7002            BSW
2035    7110            CLL RAR  /HD IN BITS 0-6
2036    1072            TAD SEC
2037    3336            DCA HEAD         /WORD 0
2040    1333            TAD CYL
2041    3337            DCA HEAD+1       /WORD 1
2042    1336            TAD HEAD
2043    1337            TAD HEAD+1
2044    1340            TAD HEAD+2
2045    7041            CIA
2046    3341            DCA HEAD+3       /WORD 3 - CHECKSUM
2047    7325            CLA STL IAC RAL  /3
2050    6517            MCR      /READ MODE
2051    7300            CLA CLL
2052    6503            SSRH     /READ SEL STATUS REG
2053    7006            RTL
2054    7006            RTL
2055    7710            SPA CLA  /READY?
2056    5252            JMP .-4  /NO
2057    7326            CLA STL RTL      /YES, 2
2060    6517            MCR      /LOAD MODE
2061    7300            CLA CLL
2062    6501            CBRH     /SELECT HEAD
2063    1210            TAD HD
2064    7106            CLL RTL
2065    7006            RTL
2066    6500            CBRL     /HEAD NO BITS 3-7
2067    7300            CLA CLL
2070    1327            TAD C5400
2071    3334            DCA COMM         /SET TO CONT. WRITE
2072    4274            JMS DSKOP        /WRITE ON DISK
2073    5600            JMP I DSK
2074    0000    DSKOP,  0
```

```
2075  7332                7332      /-6000
2076  6516                WCR       /SET WC TO MAX
2077  7300                CLA CLL
2100  1335                TAD HEADER
2101  6514                MAR       /ST OF HDR IMAGE
2102  7300                CLA CLL
2103  1334                TAD COMM
2104  6501                CBRH      /LOAD COMMAND
2105  7300                CLA CLL
2106  1072                TAD SEC
2107  7106                CLL RTL
2110  7006                RTL
2111  6500                CBRL      /SEC NO BITS3-7
2112  7300                CLA CLL
2113  7330                7330      /4000
2114  6505                AIRH      /SET WRITE REG
2115  1332                TAD C200          /DATA IN MB2
2116  6512                CSR       /GO
2117  7325                CLA STL IAC RAL  /3
2120  6517                MCR       /READ MODE
2121  7300                CLA CLL
2122  6511                ESRO      /READ ERROR STATUS
2123  5322                JMP .-1
2124  7610                SPA       /ERRORS ON DONE?
2125  7402                HLT       /YES
2126  5674                JMP I DSKOP       /NO
2127  5400    C5400,      5400
2130  1000    C1000,      1000
2131  1400    C1400,      1400
2132  0200    C200,       200
2133  0000    CYL,        0
2134  0000    COMM,       0
2135  2136    HEADER,     HEAD
2136  0000    HEAD,       0         /HEAD AND SECOR
2137  0000                0         /CYLINDER
2140  3052                -4726     /CONSTANT WC -2518
2141  0000                0         /CHECK SUM
2142  0000                0         /ADDR OF DATA - 0 MB2
                          *400
0400  7300                CLA CLL
0401  1342                TAD C3559         /ST BLOCK
0402  3071                DCA DBLK          /DATA CLOCK ON DISK
0403  4510                JMS I XSTDSK      /SET DISK CONSTANTS
0404  4450                JMS I XZRO        /ZERO OUTPUT AREA
0405  4451                JMS I XINIT       /SET ST OF BUFS
0406  3065    NXONE,      DCA HALF
0407  3074                DCA LINCT         /NO CHARS
0410  2107                ISZ UPTRG
0411  1040                TAD C3020
0412  3066                DCA DTMP          /ST OF INPUT
0413  4447                JMS I XMAG
0414  1020                1020
0415  3000                3000
0416  7000                -1000
0417  0100                100
0420  5354                JMP DOEOF         /EOF RETURN
0421  1063                TAD STOUT         /NORMAL RET
0422  1102                TAD CM50
0423  3063                DCA STOUT
0424  1063                TAD STOUT
0425  3062                DCA OUTBUF        /SET BACK NO POS
0426  1475                TAD I L3020
0427  7640                SZA CLA  /ANY DATA IN RECORD?
0430  5235                JMP DATA          /YES
```

```
0431  2067  N,      ISZ RECCT       /NO, OUTPUT BUF FULL?
0432  5206          JMP NXONE       /NO
0433  4446          JMS I XOTPUT    /YES, WRITE ON DISK
0434  5204          JMP NXONE-2
0435  4511  DATA,   JMS I XSTRT     /STORE CURSOR, NPA AND PHO
0436  1337          TAD CLIST
0437  3340          DCA RLIST       /FIELD ST LOCS
0440  2105          ISZ FLDTRG      /FIELD 1 TRIG
0441  1740  BB,     TAD I RLIST
0442  2340          ISZ RLIST
0443  7450          SNA     /DONE
0444  5326          JMP DONE        /YES
0445  3066          DCA DTMP        /NO
0446  2107          ISZ UPTRG       /SET TO UPPER CASE
0447  4453  G11,    JMS I XGETCR
0450  1070          TAD CHAR
0451  7450          SNA     /ZERO?
0452  5266          JMP ZCH  /YES
0453  1100          TAD CM77        /NO
0454  7650          CNA CLA  /SPACE?
0455  5315          JMP SPCH        /YES
0456  3106          DCA SPTRG       /NO, CLEAR TRG
0457  1107  CC,     TAD UPTRG
0460  7650          SNA CLA  /1 ST CHAR OF WD?
0461  5365          JMP CHG
0462  1070          TAD CHAR        /YES
0463  4456  XX,     JMS I XPUTCR
0464  5247          JMP GTI  /OK
0465  5326          JMP DONE        /FULL
0466  3106  ZCH,    DCA SPTRG
0467  4453          JMS I XGETCR
0470  1070          TAD CHAR
0471  1341          TAD CM31
0472  7650          SNA CLA  /FIELD CODE?
0473  5307          JMP FLD  /YES
0474  1066          TAD DTMP
0475  1343          TAD CM7441
0476  7650          SNA CLA         /DONE ALL FIELDS?
0477  5326          JMP DONE        /YES
0500  4456          JMS I XPUTCR    /NO, PUT IN BUF
0501  5303          JMP .+2  /OK
0502  7402          HLT     /FULL
0503  1070          TAD CHAR
0504  4456          JMS I XPUTCR    /PUT CHAR IN BUF
0505  5246          JMP GTI-1       /OK
0506  5326          JMP DONE        /FULL
0507  3105  FLD,    DCA FLDTRG
0510  3016          DCA SPTRG
0511  1043          TAD C77
0512  4456          JMS I XPUTCR
0513  5241          JMP BB   /O
0514  5326          JMP DONE        /FULL
0515  1105  SPCH,   TAD FLDTRG
0516  7650          SNA CLA  /FIELD 1?
0517  5246          JMP GT1-1
0520  1106          TAD SPTRG       /YES
0521  7640          SZA CLA  /FOLLOW A SPACE?
0522  5247          JMP GT1  /YES
0523  2106          ISZ SPTRG       /NO
0524  2107          ISZ UPTRG
0525  5257          JMP CC
0526  3074  DONE,   DCA LINCT
0527  1460          TAD I STORE
0530  7610          SPA     /DONE?
```

```
0531  5361              JMP DOO  /YES
0532  4456              JMS I XPUTCR     /NO
0533  5335              JMP .+2  /OK
0534  7002              HLT      /FULL
0535  2060              ISZ STORE
0536  5327              JMP DONE+1
0537  0544  CLIST,      QLIST
0540  0000  RLIST,      0
0541  7737  CM31,       -41
0542  3561  C3559,      3559     /MUST BE SET EACH TIME
0543  0337  CM7441,     -7441
0544  7200  QLIST,      7200     /NAME
0545  3303              3303     /TITLE
0546  7313              7313     /DESI
0547  7360              7360
0550  7365              7365     /HOUSE SUFFIX
0551  3336              3336     /ST
0552  3370              3370     /LOCALE
0553  0000              0
0554  7402  DOEOF,      HLT
0555  7402              HLT
0556  5206              JMP NXONE        /PROGRAM CONTINUATION
0557  4446              JMS I XOTPUT     /CONT HERE TO DUMP LAST BLOCK
0560  7402              HLT              /DONE
0561  7300  DOO,        CLA CLL
0562  1057              TAD XSTORE
0563  3060              DCA STORE
0564  5231              JMP N
0565  1070  CHG,        TAD CHAR
0566  1104              TAD CM65
0567  7700              SMA CLA          /NUMBER?
0570  5262              JMP XX-1         /YES
0571  1070              TAD CHAR         /NO
0572  1103              TAD CM32
0573  5263              JMP XX           /PUT IN LOWER CASE
                        *1000
1000  0000  STORIT,     0
1001  1100              TAD CM77
1002  3074              DCA LINCT
1003  3347              DCA NTRG
1004  1742              TAD I L3012
1005  7650              SNA CLA  /NON PUB?
1006  5230              JMP NPA  /NO
1007  4743              JMS I XCURSR     /YES, PUT CURSOR POS IN
1010  0074              74       /7
1011  0067              67       /2
1012  1344              TAD C50  /N
1013  3460              DCA I STORE
1014  2060              ISZ STORE
1015  1345              TAD C52  /P
1016  3460              DCA I STORE
1017  2060              ISZ STORE
1020  1101              TAD CM6
1021  3354              DCA C1
1022  1043  QR,         TAD C77
1023  3460              DCA I STORE
1024  2060              ISZ STORE        /PAD WITH SPACES
1025  2354              ISZ C1   /DONE?
1026  5222              JMP QR   /NO
1027  5335              JMP ENDIT        /YES
1030  4453  NPA         JMS I XGETCR     /GET 1ST NPA CHAR
```

```
1031  1070              TAD CHAR  '
1032  1077              TAD CM72
1033  7650              SNA CLA /5?
1034  5262              JMP CKON          /YES
1035  1346  PUTT,       TAD CM73          /NO
1036  3074              DCA LINCT         /-59
1037  4743              JMS I XCURSR
1040  0073              73        /6
1041  0075              75        /8
1042  1070              TAD CHAR
1043  3460              DCA I STORE
1044  2060              ISZ STORE
1045  4453              JMS I XGETCR
1046  1070              TAD CHAR
1047  3460              DCA I STORE
1050  2060              ISZ STORE
1051  4453              JMS I GETCR
1052  1070              TAD CHAR
1053  3460              DCA I STORE
1054  2060              ISZ STORE
1055  1043              TAD C77
1056  3460              DCA I STORE
1057  2060              ISZ STORE
1060  2347              ISZ NIRG
1061  5300              JMP NUM
1062  4453  CKON,       JMS I XGETCR      /GET NEXT CHAR
1063  1070              TAD CHAR
1064  1350              TAD CM66
1065  7640              SZA CLA  /1?
1066  5274              JMP REGET         /NO
1067  4453              JMS I XGETCR      /YES
1070  1070              TAD CHAR
1071  1346              TAD CM73
1072  7650              SNA CLA  /6?
1073  5300              JMP NUM  /YES, NO NPA NEEDED
1074  1040  REGET,      TAD C3020
1075  3066              DCA DIMP
1076  4453              JMS I XGETCR
1077  5235              MPT PUTT          /PUT NPA IN BUF
1100  1347  NUM,        TAC NIRG
1101  7640              SZA CLA  /CURSOR ALREADY SET?
1102  5306              JMP .+4
1103  4743              JMS I XCURSR      /NO
1104  0074              74
1105  0067              67
1106  1351              TAD C3023
1107  3066              DCA DIMP
1110  1352              TAD CM3
1111  3354              DCA C1    /DO 3 NUMBERS
1112  4453  D02,        JMS I XGETCR
1113  1070              TAD CHAR
1114  3460              DCA I STORE
1115  2060              ISZ STORE
1116  2354              ISZ C1    /DONE?
1117  5312              JMP D02   /NO
1120  3460              DCA I STORE       /YES
1121  2060              ISZ STORE
1122  1355              TAD C55
1123  3460              DCA I STORE       /HYPHEN
1124  2060              ISZ STORE
1125  1353              TAD CM4
1126  3354              DCA C1    /DO 4 NUMBERS
1127  4453  D03,        JMS I XGETCR
1130  1070              TAD CHAR
```

```
1131  3460              DCA I STORE
1132  2060              ISZ STORE
1133  2354              ISZ C1    /DONE?
1134  5327              JMP DO3   /NO
1135  1076  ENDIT,      TAD C4000
1136  3460              DCA I STORE       TERM WITH NO ONE
1137  1057              TAD XSTORE
1140  3060              DCA STORE
1141  5600              JMP I STORIT
1142  3011  L3012,      3011
1143  2200  XCURSR,     CURSOR
1144  0050  C50,        50
1145  0052  C52,        52
1146  7765  CM73,       -73
1147  0000  NIRG,       0
1150  7712  CM66,       -66
1151  7024  C3023,      7024
1152  7775  CM3,        -3
1153  7774  CM4,        -4
1154  0000  C1,         0
1155  0055  C55,        55
                        *2200
2200  0000  CURSOR,     0
2201  1600              TAD I CURSOR
2202  3230              DCA POS1
2203  2200              ISZ CURSOR
2204  1600              TAD I CURSOR
2205  3231              DCA POS2
2206  2200              ISZ CURSOR
2207  1101              TAD CM6
2210  3223              DCA CTTR
2211  1222              TAD XROW
2212  3232              DCA ROW
2213  1632  DO1,        TAD I ROW
2214  3460              DCA I STORE
2215  2232              ISZ ROW
2216  2060              ISZ STORE
2217  2223              ISZ CTTR          /DOWN?
2220  5213              JMP DO1  /NO
2221  5600              JMP I CURSOR      /YES
2222  2224  XROW,       ROWR
2223  0000  CTTR,       0
2224  0000  ROWR,       0
2225  0043              43
2226  0000              0
2227  0044              44
2230  0000  POS1,       0
2231  0000  POS2,       0
2232  0000  ROW,        0
                        *3000
                /XDIVID SINGLE PRECISION DIVIDE SUBROUTINE
                /       BOUNDS OF DIVIDEND: 0-7777
                /       BOUNDS OF DIVISOR: 1-3777
                /CALL   JMS I XDIVID
                /       (DIVIDEND)
                /       (DIVISOR)
                /       REMAINDER RETURNED HERE
                /       CONTROL RESUMES HERE WITH QUOTIENT IN AC
```

```
5000  0000  DIVIDE,   0
5001  7100            CLL
5002  3253            DCA HDIV
5003  1600            TAD I DIVIDE
5004  2200            ISZ DIVIDE
5005  3254            DCA LDIV
5006  1600            TAD I DIVIDE
5007  2200            ISZ DIVIDE
5010  7041            CIA
5011  3255            DCA DIV
5012  1253            TAD HDIV
5013  7640            SZA CLA
5014  5235            JMP DV2
5015  1254            TAD LDIV
5016  1255            TAD DIV
5017  7620            SNL CLA  /DIV<DIVISOR?
5020  5247            JMP DV4  /YES
5021  7300            CLA CLL
5022  1256            TAD CM15
5023  3257            DCA DIVCT
5024  5235            JMP DV2
5025  1253  DV3,      TAD HDIV
5026  7004            RAL
5027  3253            DCA HDIV
5030  1253            TAD HDIV
5031  1255            TAD DIV
5032  7430            SZL
5033  3253            DCA HDIV
5034  7200            CLA
5035  1254  DV2,      TAD LDIV
5036  7004            RAL
5037  3254            DCA LDIV
5040  2257            ISZ DIVCT
5041  5225            JMP DV3
5042  1253            TAD HDIV
5043  3600            DCA I DIVIDE
5044  2200            ISZ DIVIDE
5045  1254            TAD LDIV
5046  5600            JMP I DIVIDE
5047  1254  DV4,      TAD LDIV        /QUOTIENT=0, REMAINDER
5050  3600            DCA I DIVIDE    /=DIVIDEND
5051  2200            ISZ DIVIDE
5052  5600            JMP I DIVIDE
5053  0000  HDIV,     0
5054  0000  LDIV,     0
5055  0000  DIV,      0
5056  7763  CM15,     -15
5057  0000  DIVCT,    0

                      *200
                /FORMAT:          JMS I XMAG
                /                 COMMAND
                /                 ADDRESS
                /                 WORD COUNT
                /                 EXTENSION REGISTER
                /       RETURN:   EOF
                /       RETURN:   NORMAL
```

```
0200  0000  MAG,     0
0201  1335           TAD MCM12
0202  3347           DCA REVCNT      /SET 10 RETRIES
0203  1600           TAD I MAG
0204  2200           ISZ MAC
0205  3350           DCA COM         /GET COMMAND
0206  7240           CLA CMA
0207  1600           TAD I MAG
0210  3351           DCA CADDR       /CET CURRENT ADDRESS
0212  1600           TAD I MAG
0213  2200           ISZ MAC
0214  3352           DCA WRDCNT      /GET WORD COUNT
0215  1600           TAD I MAG
0216  2200           ISZ MAG
0217  3346           DCA MADCOM
0221  1350  RETRY,   TAD COM
0222  1346           TAD MADCOM
0223  4322           JMS SETCOM      /SET CONTROLLER FOR FUNCTION
0224  1352           TAD WRDCNT
0225  3032           DCA 32
0226  1351           TAD CADDR
0227  3033           DCA 33          /SET WORD COUNT & CURRENT ADDRESS
0230  1353           TAD COMEX
0231  6717           6717
0232  7300           CLA CLL         /SET EXT REG
0233  4330           JMS MAGOP       /PERFORM MAGTAPE FUNCTION
0234  6706           6706
0235  7421           MQL             /STORE STATUS IN MQ
0236  7501           MQA
0237  0336           AND MC6774
0240  7450           SNA
0241  5271           JMP MOK         /NO ERRORS
0242  0337           AND MC7677
0243  7450           SNA
0244  5273           JMP MEOF        /EOF
0245  0340           AND MC3543
0246  7640           SZA CLA
0247  7402           HLT             /BAD REC OR OFFLINE
0250  2347  PAR,     ISZ REVCNT      /PARITY
0251  5253           JMP .+2         /RETRY
0252  7402           HLT             /RETRY FAILURE
0253  1350           TAD COM
0254  0343           AND MC70
0255  1344           TAD MCM40
0256  7650           SNA CLA         /FAILURE ON MT?
0257  1341           TAD MC100       /YES
0260  3346           DCA MADCOM
0261  1350           TAD COM
0262  0342           AND MC7000
0263  1343           TAD MC70
0264  4322           JMS SETCOM      /SET CONTROLLER FOR BACKSPACE
0265  7240           CLA CMA
0266  3032           DCA 32          /WC = RECS BACKSPACE
0267  4330           JMS MAGOP       /PERFORM BACKSPACE
0270  5221           JMP RETRY       /AND TRY AGAIN
0271  2200  MOK,     ISZ MAG
0272  5600           JMP I MAG       /NORMAL EXIT
0273  1350  MEOF,    TAD COM
0274  0343           AND MC70
0275  1344           TAD MCM40
```

```
0276  7650              SNA CLA          /EOF ON WRITE?
0277  5250              JMP PAR          /YES - PARITY PROBLEM
0300  7000              NOP
0301  5600              JMP I MAG        /EXIT EOF
0302  0000  REWIND,     0
0303  1350              TAD COM
0304  0342              AND MC7000
0305  1345              TAD MC10
0306  4322              JMS SETCOM       /SET CONTROLLER FOR REWIND
0307  6722              6722             /EXECUTE REWIND
0310  1350              TAD COM
0311  0342              AND MC7000
0312  7106              CLL RTL
0313  7006              RTL              /LOAD CONT W/ NEXT MAG
0314  1355              TAD NEXM
0315  3354              DCA NEXCOM
0316  1764              TAD I NEXCOM
0317  6716              6716
0320  7300              CLA CLL
0321  5702              JMP I REWIND
0322  0000  SETCOM,     0
0323  6711              6711
0324  5323              JMP .-1
0325  6716              6716
0326  7300              CLA CLL
0327  5722              JMP I SETCOM
0330  0000  MAGOP,      0
0331  6722              6722
0332  6701              6701
0333  5332              JMP .-1
0334  5730              JMP I MAGOP
0335  7766  MCM12,      -12
0336  6774  MC6774,     6774
0337  7677  MC7677,     7677
0340  3543  MC3543,     3543
0341  0100  MC100,      100
0342  7000  MC7000,     7000
0343  0070  MC70,       70
0344  7740  MCM40,      -40
0345  0010  MC10,       10
0346  0000  MADCOM,     0
0347  0000  REVCNT,     0
0350  0000  COM,        0
0351  0000  CADDR,      0
0352  0000  WRDCNT,     0
0353  0000  COMEX,      0
0354  0000  NEXCOM,     0
0355  0356  NEXM,       NEXTM
0356  1020  NEXTM,      1020      /0
0357  0040              40        /1
0360  0040              40        /2
0361  0040              40        /3
```

```
                        *2700
2700    0000    OCDEC,  0
2701    7600            CLA CLL
2702    1700            TAD I OCDEC
2703    2300            ISZ OCDEC
2704    3355            DCA UDHIGH
2705    1700            TAD I OCDEC
2706    2300            ISZ OCDEC
2707    3356            DCA UDLOW
2710    1351            TAD UDLOOP
2711    3354            DCA UDCNT
2712    1352            TAD UNADDR
2713    3364            DCA UDPTR
2714    3361            DCA UDBOX
2715    1764    UDARND  TAD I UDPTR
2716    2364            ISZ UDPTR
2717    3357            DCA UDHSUB
2720    1764            TAD I UDPTR
2721    2364            ISZ UDPTR
2722    3360            DCA UDLSUB
2723    7100    UDDO,   CLL
2724    1360            TAD UDLSUB
2725    1356            TAD UDLOW
2726    3362            DCA UDTEML
2727    7004            RAL
2730    1357            TAD UDHSUB
2731    1355            TAD UDHIGH
2732    7420            SNL
2733    5341            JMP UDOUT
2734    2361            ISZ UDBOX
2735    3355            DCA UDHIGH
2736    1362            TAD UDTEML
2737    3356            DCA UDLOW
2740    5323            JMP UDDO
2741    7200    UDOUT,  CLA
2742    1361            TAD UDBOX
2743    3700            DCA I OCDEC
2744    2300            ISZ OCDEC
2745    3361            DCA UDBOX
2746    2354            ISZ UDCNT
2747    5315            JMP UDARND
2750    5700            JMP I OCDEC
2751    7770    UDLOOP, -10
2752    2765    UNADDR, UDCON1
2753    0260    UDTWO,  260
2754    0000    UDCNT,  0
2755    0000    UDHIGH, 0
2756    0000    UDLOW   0
2757    0000    UDHSUB, 0
2760    0000    UDLSUB, 0
2761    0000    UDBOX,  0
2762    0000    UDTEML, 0
2763    0000    UDGET,  0
2764    0000    UDPTR,  0
2765    3166    UDCON1, 3166    /POWERS OF TEN
2766    4600            4600
2767    7413            7413
2770    6700            6700
2771    7747            7747
2772    4540            4540
2773    7775            7775
2774    4360            4360
2775    7777            7777
2776    6030            6030
2777    7777            7777
3000    7634            7634
3001    7777            7777
3002    7766            7766
3003    7777            7777
3004    7777            7777
```

```
                        *1700
1700    0000    SETDSK, 0       /SET CONSTANT DISK REGISTERS
1701    1315            TAD C3000
1702    6517            MCR     /SELECT CONTROLLER 0
1703    7326            CLA STL RTL     /2
1704    6517            MCR     /LOAD MODE
1705    7300            CLA CLL
1706    7333            7333    /6000
1707    6501            CBRH    /LOAD DRIVE
1710    7300            CLA CLL
1711    1073            TAD DRIVE
1712    6500            CBRL
1713    7300            CLA CLL
1714    5700            JMP I SETDSK
1715    3000    C3000,  3000
                        *2600
2600    0000    GETCR,  0
2601    1066            TAD DTMP
2602    7600            SMA             /DATA IN 1ST HALF?
2603    5213            JMP SECN        /NO
2604    0227            AND C3777       /YES
2605    3066            DCA DTMP
2606    1466            TAD I DTMP
2607    7002            BSW
2610    0043            AND C77
2611    3070            DCA CHAR
2612    5225            JMP EXIT
2613    3066    SECN,   DCA DTMP
2614    1466            TAD I DTMP
2615    0043            AND C77
2616    3070            DCA CHAR
2617    1076            TAD C4000
2620    7421            MQL             /SET UP TRIGGER
2621    2066            ISZ DTMP
2622    1066            TAD DMTP
2623    7601            MQA
2624    3066            DCA DTMP
2625    7000    EXIT,   NOP
2626    5600            JMP I GETCR
2627    3777    C3777,  3777

AG1     1406
AG2     1602
BB      0441
CADDR   0351
CC      0457
CHAR    0070
CHG     0565
CHR     1643
CKON    1062
CLIST   0537
CM123   0041
CM15    5056
CM3     1152
CM31    0541
CM32    0103
CM4     1153
CM4726  1414
CM50    0102
CM5763  1566
CM6     0101
CM65    0104
CM66    1150
```

CM72	0077
CM73	1146
CM74	1441
CM7441	0543
CM77	0100
COM	0350
COMEX	0353
COMM	2134
CTR	1415
CTTR	2223
CURSOR	2200
CYC	2133
C1	1154
C1000	2130
C1400	2131
C200	2132
C3000	1715
C3020	0040
C3023	1151
C3559	0542
C3777	2627
C4000	0076
C4020	0042
C4726	1440
C50	1144
C52	1145
C5400	2127
C55	1155
C77	0043
DATA	0435
DBLK	0071
DDBLK	1555
DIV	5055
DIVCT	5057
DIVIDE	5000
DIVR	2006
DOEOF	0554
DONE	0526
DO0	0561
DO1	2213
DO2	1112
DO3	1127
DRIVE	0073
DSK	2000
DSKOP	2074
DUMP	0066
DV2	5035
DV3	5025
DV4	5047
END	0061
ENDIT	1135
EXIT	2625
FLD	0507
FLDIRG	0105
GETCR	2600
GT1	0447
HALF	0065
HD	2010
HDIV	5053
HEAD	2136
HEADER	2135

INIT	1430
LDIV	5054
LINCT	0074
LINE	0064
L2	0044
L3	0045
L3012	1142
L3020	0075
MADCOM	0346
MAG	0200
MAGOR	0330
MCM12	0335
MCM40	0344
MC10	0345
MC100	0341
MC3543	0340
MC6774	0336
MC70	0343
MC7000	0342
MC7677	0337
MEOF	0273
MOK	0271
N	0431
NEXCOM	0354
NEXM	0355
NEXTM	0356
NNN	0024
NPA	1030
NTRG	1147
NUM	1100
NXONE	0406
OCDEC	2700
OTPUT	1550
OUTBUF	0062
PAR	0250
POS1	2230
POS2	2231
PUTCR	1610
PUTT	1035
PUTWD	1600
QLIST	0544
QQ	1644
QR	1022
RECCT	0067
REGET	1074
RETRY	0221
REVCNT	0347
REWIND	0302
RLIST	0540
ROW	2232
ROWR	2224
SEC	0072
SECHLF	1634
SECN	2613
SETCOM	0322
SETDSK	1700
SPCH	0515
SPTRG	0106
STORE	0060
STORIT	1000
STOUT	0063

```
16        1627
UDADDR    2752
UDARND    2715
UDBOK     2761
UDCNT     2754
UDCON1    2765
UDDO      2723
UDGET     2763
UDHGH     2755
UDHSUB    2757
UDLOOP    2751
UDLOW     2756
UDLSUB    2760
UDOUT     2741
UDPTR     2764
UDTEML    2762
UDTWO     2753
UPTRG     0107
WORD      0112
WRDCNT    0352
XCURSR    1143
XDIVID    0052
XDSK      1567
XGETCR    0053
XINIT     0051
XMAG      0047
XOTPUT    0046
XPUTCR    0056
XPUTWD    0054
XREWD     0055
XROW      2222
XSTDSK    0110
XSTORE    0057
XSTRT     0111
XX        0463
XZRO      0050
ZCH       0466
ZRO       1400
```

Operating Instructions for the above listing Program No. 5 are as follows:

1. Load Program #5 into MBO.
2. Put input tape on unit 1.
3. Set the following parameters for unit being processed:

   LOCATION

   72 = sector (0 or 14)

   73 = drive (0 or 20)

   542 = starting block # See Sheet

   1566 = block limit

4. 400 load, clear, and continue. Do not clear after starting program.

5. Halt at 555 indicates input eof has been read. Load and examine locations 3000-3020. They should all be zero. If not, load and continue at location 20. If these locations are zeroed, go to instruction 6.

6. (a) If no more input tapes: load and examine loc 63. If loc 63 ≠ 4626, 557 load and continue to dump final data block. Note contents of locs 63 and 1555. Manually rewind tape. If loc 63 = 4626, note contents of locs 63 and 1555 and rewind tape manually.

   (b) If more input tapes: Be sure next tape is on unit. 556 load and continue.

Specifications

1. Input: a. NYT/DRC tape, 1600 bpi, special core dump mode.

   b. Input buffer: 3000-4000 MB0.

2. Output: Diva disk from 0 - 4726 MB2.

   60 - 80 character (40 location) data records per block.

HALT LIST

Halt Location	Reason for Halt	Recovery Procedure
247	Bad tape or offline	Check drives
252	Retry failure	If loc 350=1020, 20 load and continue. If loc 350=240, 253 load and continue.

Halt Location	Reason for Halt	Recorvery Procedure
502	Buffer overflow on special character. (zero already in buffer)	Load and examine loc 62. This gives the position following the zero, using the standard 4000 trigger. Locate the zero in mb2, and replace it with a space. Return to mb0. 526 load and continue.
534	Buffer overflow on tel. number	Get programming assistance.
554	Input eof.	See instruction 6.
560	Program completed	
1564	Disk limit overflow	Abort, more than 30,000 frames
1642	Output buffer overflow	Abort
2125	Disk error	Retry

UNIT	STARTING BLOCK (LOCATION 542)	BLOCK LIMIT (LOCATION 1566)	
L,E	1057 (559)	7705 (-59)	SEC 0
N, MI-MZ	3027 (1559)	5735 (-1059)	
W, BR-BZ	4777 (2559)	3765 (-2059)	
CR, Q, B	6747 (3559)	2015 (-3059)	
P, I, Z	1057	7705	SEC 14
C - CQ	3027	5735	
K, J, U	4777	3765	
R, V	6747	2015	
X, Y, D	1057	7705	SEC 0
G	3027	5735	
H	4777	3765	
F	6747	2015	
S-SN	1057	7705	SEC 14
A, T	3027	5735	
SO-SZ, O	4777	3765	
M	6747	2015	

C3. Detailed Description of Program No. 3

Program No. 3 is shown in block form at FIGURE 44 with subroutines utilized therein detailed in FIGURES 45-52. An explicit listing of the assembly level source program language for Program No. 3 and all related subroutines follows. It will be noted that the main program is shown as comprising sections I, II, III and IV in FIGURE 44. These program sections correspond to specific listing statements.

Section	Instruction Statement
I	0200-0202
II	0203-0204; 0216-0223
III	0205-0206
IV	0207-0215

These four program sections are briefly functionally described below as an introduction to the explicit program listing:

I. Initialize.

A. "REC" is used as a record pointer for the output block. Twleve is the blocking factor. (REC 0-11)

B. "BUF" zeroes the output block area.

II. Read input record.

A. "READ" goes to XMAG to physically read the input tape and indicates whether a tape mark has been reached by its returning position. If a normal exit occurs the program continues to part III (MX processing).

B. If a tape mark is encountered the input tape is rewound and the program outputs the last MX block

if there are any records in it and halts. If the operator desires to manually continue, a tape mark will be placed on the output tape (it is also rewound) and again the program halts (processing is complete).

III. Process MX Record. "XMX" translates the input record into MX (PIC matrix) format and stores it in the proper record of the output block (determined by "REC").

IV. Output

A. Set record pointer "REC" to next record.

B. If output block is full, write output block and go to I, otherwise go to II.

The XMAG subroutine is not shown in the FIGURES but is described below:

XMAG: general subroutine to drive the magnetic tape transport.

Format: JMS I XMAG
(command) *2
(current address)
(word count)
(extension register) *3
control resumes here if tape mark encountered
control resumes here normally

*2. command bits:
0 → formatter select
1,2 → unit select
3-5 → N/A
6-8 → tape command
9-n → N/A

*3. extension bits
0-4 → N/A
5 → special core dump
6-9 → N/A
10-11 → memory bank

Program No. 3

```
                BSW=7002
                MQC=7421
                MQA=7501
                SWP=7521
                CEDO=6201
                CDF1=6211
                CDF2=6221
                @MCR=6517
                DCBRH=6501
                DCBRL=6500
                DSSRH=6503
                DSSRL=6502
                DESRO=6511
                DESR1=6515
                DWCR=6516
                DDMAR=6514
                DCSR=6512
                DAIRL=6504
                DAIRH=6505
                DUSRL=6506
                DUSRH=6507
                GTF-6004
                RTF=6005
                RXF=6244
                RIB=6234
                SRQ=6003

                FIXTAB
                                *20
0020    0000    MTMP,           0
0021    0000    MADD,           0
0022    0000    GLOC,           0
0023    0000    PADD,           0
0024    0000    MCT,            0
0025    0000    MXLOC,          0
0026    0000    PLIM,           0
0027    0044    MFAC,           BITS-CADDS
0030    1000    CADD,           CADDS
0031    0266    XGTCHR,         GTCHR

                PAGE

                /NYT MX         PROGRAM

0200    7300    ST1,            CLA CLL
0201    3206                    DCA REC
0202    4224                    JMS BUF
0203    4235    ST2,            JMS READ        /INPUT DRC
0204    5216                    JMP ST3  /EOF
0205    4722                    JMS I XMX       /PROCESS TO MX
0206    0000    REC,            0       /AT REC X
0207    1206                    TAD REC
0210    2206                    ISZ REC
0211    1317                    TAD CM13
0212    7640                    SZA CLA  /LAST REC IN BLK?
0213    5203                    JMP ST2  /NO
0214    4256                    JMS WRITE       /OUTPUT BLK
0215    5200                    JMP ST1
```

```
0216  1206  ST3,      TAD REC
0217  7640            SZA CLA  /ANY RECS IN OUTPUT BLK?
0220  4256            JMS WRITE        /YES - OUTPUT BLK
0221  7402            HLT              /STOP AFTER INPUT TAPE PROCESSED
0222  4246            JMS WEOF         /OUTPUT EOF
0223  7402            HLT      /PROGRAM COMPLETED

0224  0000  BUF,       0
0225  1315            TAD C4777
0226  3010            DCA 10
0227  1316            TAD CM2000
0230  3321            DCA CNT
0231  3410            DCA I 10
0232  2321            ISZ CNT
0233  5231            JMP .-2
0234  5624            JMP I BUF

0235  0000  READ,      0
0236  4723            JMS I XMAG       /INPUT FIELDED DRC RED
0237  1020            1020     /COM,ADR,WC,EXT REG
0240  3000            3000
0241  7000            -1000
0242  0100            100
0243  5635            JMP I READ       /EOF (AUTO REWIND)
0244  2235            ISZ READ
0245  5635            JMP I READ

0246  0000  WEOF,      0
0247  4723            JMS I XMAG       /OUTPUT EOF
0250  0050            50
0251  0000            0
0252  0000            0
0253  0000            0
0254  7300            CLA CLL
0255  5646            JMP I WEOF

0256  0000  WRITE,     0
0257  4723            JMS I XMAG       /OUTPUT MX BLK
0260  0040            40       /COM,ADR,WC,EXT REG
0261  5000            5000
0262  6000            -2000
0263  0100            100
0264  7402            HLT
0265  5656            JMP I WRITE

0266  0000  GTCHR,     0
0267  3320            DCA GGLOC
0270  1320            TAD GGLOC
0271  7500            SMA      /1ST HLF?
0272  5303            JMP G2   /NO
0273  0312            AND C3777
0274  3320            DCA GGLOC
0275  1720            TAD I GGLOC
0276  7002            BSW
0277  0313            AND C77
0300  7421            MQL
0301  1320            TAD GGLOC
0302  5666            JMP I GTCHR
```

```
0303  7300  C2,       CLA CLL
0304  1720            TAD I GGLOC
0305  0313            AND C77
0306  7421            MQL
0307  1320            TAD GGLOC
0310  1314            TAD C4001
0311  5666            JMP I GTCHR

0312  3777  C3777,    3777
0313  0077  C77,      77
0314  4001  C4001,    4001
0315  4777  C4777,    4777
0316  6000  CM2000,   -2000
0317  7765  CM13,     -13
0320  0000  GGLOC,    0
0321  0400  XMX,      MX
0323  0600  XMAG,     MAG

            PAGE

            /PROCESS NXT FIELDS TO MX

0400  0000  MX,       0
0401  1600            TAD I MX
0402  2200            ISZ MX
0403  7002            BSW
0404  3021            DCA MADD          /START OF REC IN BLK
0405  1760            TAD I M3033
0406  7650            SNA CLA
0407  5600            JMP I MX          /BLANK REC
0410  1371            TAD MC5074
0411  1021            TAD MADD
0412  3372            DCA MBTMP
0413  1773            TAD I ML3012
0414  7450            SNA
0415  5222            JMP MBIT1
0416  1374            TAD MCM10
0417  7650            SNA CLA
0420  1375            TAD MC2000        /PROF: SET BITS (6000)
0421  1375            TAD MC2000        /BUS: SET BIT 0 (4000)
0422  1375  MBIT1,    TAD MC2000        /RES: SET BIT 1 (2000)
0423  3772            DCA I MBTMP       /CATEGORY BIT - MX
0424  1356            TAD MC4000        /ORDINARILY STRT WITH 2ND CHR
0425  1357  M2,       TAD MC3034        /S:     B,C,M,S, STRT WITH 3RD CHR
0426  3022            DCA GLOC

0427  1367  MFLD1,    TAD MCM22
0430  3026            DCA PLIM          /ONLY ALLOW 6 MX CHRS:
0431  3023            DCA PADD          /SET 1ST CHR POS OF WD
0432  4265            JMS MXCHR         /GET CHRS, PROCESS UNIT
0433  5235            JMP MFLD2         /SPACE OR SPECIAL
0434  7000            NOP       /FIELD DEL

0435  1021  MFLD2,    TAD MADD
0436  1360            TAD MC22
0437  3021            DCA MADD          /FIELD STRT
0440  1361            TAD MCM2
0441  3024            DCA MCT  /SET 2 FLDS
0442  3023  MFLD21,   DCA PADD          /SET 1ST CHR POS OF WD
0443  4265  MFLD22,   JMS MXCHR         /GET CHRS, PROCESS UNTIL
0444  5242            JMP MFLD21        /SPACE OR SPECIAL
0445  2024            ISZ MCT  /FIELD DEL
0446  5242            JMP MFLD 21
```

```
0447  1370    MFLD3,    TAD MCM14
0450  3026              DCA PLIM          /ONLY ALLOW 4 MX HRS:
0451  1021              TAD MADD
0452  1360              TAD MC22
0453  3021              DCA MADD          /FIELD STRT
0454  3023    MFLD31,   DCA PADD          /1ST CHR POS OF WD
0455  4265              JMS MXCHR         /GET CHRS, PROCESS UNTIL
0456  5254              JMP MFLD31        /SPACE OR SPECIAL
0457  1362              TAD MC3336        /FIELD DEL
0460  3022              DCA GLOC
0461  3023    MFLD32,   DCA PADD          /1ST CHR POS OF WD
0462  4265              JMS MXCHR         /GET CHRS, PROCESS UNTIL
0463  5261              JMP MFLD32        /SPACE OR SPECIAL
0464  5261              JMP MFLD32        /FIELD DEL

0465  0000    MXCHR,    0
0466  1022              TAD GLOC
0467  4431              JMS I XGTCHR
0470  3022              DCA GLOC
0471  7501              MQA
0472  7450              SNA
0473  5304              JMP MDEL          /0:     DEL
0474  1363              TAD MCM33
0475  7610              SPA
0476  5302              JMP MSPA          /1-32:  SPACE
0477  1364              TAD MCM44
0500  7710              SPA CLA
0501  5316              JMP MCHR          /32-76: UCA;#
0502  7300    MSPA,     CLA CLL
0503  5665              JMP I MXCHR
0504  1022    MDEL,     TAD GLOC
0505  4431              JMS I XGTCHR
0506  3022              DCA GLOC
0507  7501              MQA
0510  7450              SNA
0511  1600              JMP I MX          /END OF REC
0512  1365              TAD MCM41
0513  7650              SNA CLA
0514  2265              ISZ MXCHR         /FIELD DEL
0515  5665              JMP I MXCHR       /SPECIAL

0516  1023    MCHR,     TAD PADD
0517  1026              TAD PLIM
0520  7650              SNA CLA  /TOO MANY CHRS IN WORD?
0521  5266              JMP MXCHR+1       /YES - IGNORE
0522  7701              CLA MQA
0523  1363              TAD MCM33
0524  1030              TAD CADD
0525  3020              DCA MTMP
0526  1021              TAD MADD          /MX FLD STRT +
0527  1023              TAD PADD          /POS FACTOR +
0530  1420              TAD I MTMP        /CHR STRT =
0531  1347              TAD MC5000
0532  3025              DCA MXLOC         /LOC OF MX BIT
0533  1020              TAD MTMP
0534  1027              TAD MFAC
0535  3020              DCA MTMP
0536  1420              TAD I MTMP        /BIT
0537  7421              MQL
0540  1425              TAD I MXLOC
0541  7501              MQA       /OF BIT INTO MX
0542  3425              DCA I MXLOC
0543  1023              TAD PADD
0544  1366              TAD MC3   /SET NEXT CHR POS
0545  3023              DCA PAD
0546  5266              JMP MXCHR+1
```

```
0547    5000    MC5000,     5000
0550    3033    M3033,      3033
0551    0077    M77,        77
0552    7744    MCM34,      -34
0553    7777    MCM1        -1
0554    7766    MMCM12,     -12
0555    7772    MCM6,       -6
0556    4000    MC4000,     4000
0557    3034    MC3034,     3034
0560    0022    MC22,       22
0561    7776    MCM2,       -2
0562    3336    MC3336,     3336
0563    7745    MCM33,      -33
0564    7734    MCM44,      -44
0565    7737    MCM41,      -41
0566    0003    MC3,        3
0567    7756    MCM22,      -22
0570    7764    MCM14,      -14
0571    5074    MC5074,     5074
0572    0000    MSTMP,      0
0573    3012    ML3012,     3012
0574    7770    MCM10,      -10
0575    2000    MC2000,     2000

                PAGE

                /FORMAT:            JMS I XMAG
                /                   COMMAND
                /                   ADDRESS
                /                   WORD COUNT
                /                   EXTENSION REGISTER
                /                   RETURN: EOF
                /                   RETURN: NORMAL

0600    0000    MAG,        0
0601    1337                TAD MCM12
0602    3351                DCA REVCNT      /SET 10 RETRIES
0603    1600                TAD I MAG
0604    2200                ISZ MAG
0605    3352                DCA COM   GET COMMAND
0606    7240                CLA CMA
0607    1600                TAD I MAG
0610    2200                ISZ MAG
0611    3353                DCA CADDR       /GET CURRENT ADDRESS
0612    1600                TAD I MAG
0613    2200                ISZ MAG
0614    3354                DCA WRDCNT      /GET WORD COUNT
0615    1600                TAD I MAG
0616    2200                ISZ MAG
0617    3355                DCA COMEX       /GET EXT REGISTER
0620    3350                DCA MADCOM
0621    1352    RETRY,      TAD COM
0622    1350                TAD MADCOM
0623    4322                JMS SETCOM      /SET CONTROLLER FOR FUNCTION
0624    1354                TAD WRDCNT
0625    3032                DCA 32
0626    1353                TAD CADDR
0627    3033                DCA 33   /SET WORD COUNT & CURRENT ADDRESS
0630    1355                TAD COMEX
```

```
0631  6717              6717
0632  7300              CLA CLL  /SET EXT REG
0633  4330              JMS MAGOP        /PERFORM MAGTAPE FUNTION
0634  6706              6706
0635  7421              MQL      /STORE STATUS IN MQ
0636  7501              MQA
0637  0340              AND MC6774
0640  7450              SNA
0641  5271              JMP MOK  /NO ERRORS
0642  0341              AND MC7677
0643  7450              SNA
0644  5273              JMP MEOF         /EOF
0645  0342              AND MC3543
0646  7540              SZA CLA
0647  7402              HLT      /BAD REC ON LINE
0650  2351  PAR,        ISZ REVCNT       PARMTY
0651  5253              JMP .+2  /RETRY
0652  7402              HLT      /RETRY FAILURE
0653  1352              TAD COM
0654  0345              AND MC70
0655  1346              TAD MCM40
0656  7650              SNA CLA  /FAILURE ON MT?
0657  1343              TAD MC100        /YES
0660  3350              DCA MADCOM
0661  1352              TAD COM
0662  0344              AND MC7000
0663  1345              TAD MC70
0664  4322              JMS SETCOM       /SET CONTROLLER FOR BACKSPACE
0665  7240              CLA CMA
0666  3032              DCA 32   WC = RECS BACKSPACED
0667  4330              JMS MAGOP        /PERFORM BACKSPACE
0670  5221              JMP RETRY        /AND TRY AGAIN
0671  2200  MOK,        ISZ MAG
0672  5600              JMP I MAG        /NORXAL EXIT
0673  1352  MEOF,       TAD COM
0674  0345              AND MC70
0675  1346              TAD MCM40
0676  7650              SNA CLA  /EOF ON WRITE EOF?
0677  5600              JMP I MAG        /YES - IGNORE REWIND
0700  4302              JMS REWIND       /REWIND UNIT
0701  5600              JMP I MAG        /EXIT EOF
0702  0000  REWIND,     0
0703  1352              TAD COM
0704  0344              AND MC7000
0705  1347              TAD MC10
0706  4322              JMS SETCOM       /SET CONTROLLER FOR REWIND
0707  6722              6722     /EXECUTE REWIND
0710  1352              TAD COM
0711  0344              AND MC7000
0712  7106              CLL RTL
0713  7006              RTL      /LOAD CONT W/ NEXT MAG
0714  1357              TAD NEXM
0715  3356              DCA NEXCOM
0716  1756              TAD I NEXCOM
0717  6716              6716
0720  7300              CLA CLL
0721  5702              JMP I REWIND
0722  0000  SETCOM,     0
0723  6711              6711
0724  5323              JMP .-1
0725  6716              6716
0726  7300              CLA CLL
0727  5722              JMP I SETCOM
0730  0000  MAGOP,      0
```

```
0731    6721                    6721
0732    5331                    JMP .-1
0733    6722                    6722
0734    6701                    6701
0735    5334                    JMP .-1
0736    5780                    JMP I MAGOP
0737    7766    MCM12,          -12
0740    6774    MC6774,         6774
0741    7677    MC7677,         7677
0742    3543    MC3543,         3543
0743    0100    MC100,          100
0744    7000    MC7000,         7000
0745    0070    MC70            70
0746    7730    MCM40,          -50
0747    0010    MC10,           10
0750    0000    MADCOM,         0
0751    0000    REVCNT,         0
0752    0000    COM,            0
0753    0000    CADDR,          0
0754    0000    WRDCNT,         0
0755    0000    COMEX,          0
0756    0000    NEXCOM,         0
0757    0760    NEXM,           NEXTM
0760    1020    NEXTM,          1020    /0
0761    0040                    40      /1
0762    0040                    40      /2
0673    0040                    40      /3

                PAGE

                /DRC EQUIVALENTS TABLES

1000    0000    CADDS,          0       /A-->L
1001    0000                    0;
1002    0000    0;
1003    0000    0;
1004    0000    0;
1005    0000    0;
1006    0000    0;
1007    0000    0;
1010    0000    0;
1011    0000    0;
1012    0000    0;
1013    0000    0
1014    0001                    1       /M-->X
1015    0001                    1;
1016    0001    1;
1017    0001    1;
1020    0001    1;
1021    0001    1;
1022    0001    1;
1023    0001    1;
1024    0001    1;
1025    0001    1;
1026    0001    1;
1027    0001    1
1030    0002                    2       /Y,Z
1031    0002                    2
1032    0014                    14      /0-->9
1033    0014                    14;
1034    0014    14;
1035    0014    14;
```

```
1036    0014    14;
1037    0014    14;
1040    0014    14;
0041    0014    14;
1042    0014    14;
1043    0014    14

1044    4000    BITS,       4000        /A-->L
1045    2000                2000;
1046    1000    1000;
1047    0400    400;
1050    0200    200;
1051    0100    100;
1052    0040    40;
1053    0020    20;
1054    0010    10;
1055    0004    4;
1056    0002    2;
1057    0001    1
1060    4000                4000        /M-->X
1061    2000                2000;
1062    1000    1000;
1063    0400    400;
1064    0200    200;
1065    0100    100;
1066    0040    40;
1067    0020    20;
1070    0010    10;
1071    0004    4;
1072    0002    2;
1073    0001    1
1074    4000                4000        /Y,Z
1075    2000                2000
1076    4000                4000        /0-->9
1077    2000                2000;
1100    1000    1000;
1101    0400    400;
1102    0200    200;
1103    0100    100;
1104    0040    40;
1105    0020    20;
1106    0010    10;
1107    0004    4

                PAGE

BITS    1044
BUF     0224
CADD    0030
CADDR   0753
CADD8   1000
CM13    0317
CM2000  0316
CNT     0321
COM     0752
COMEX   0755
C3777   0312
C4001   0314
C4777   0315
C77     0313
GGLOC   0320
GLOC    0011
```

GTCHR	3266
G2	0303
MADCOM	0750
MADD	0021
MAG	0600
MAGOP	0730
MBIT1	0422
MBTMP	0572
MCHR	0516
MCM1	0553
MCM10	0574
MCM12	0737
MCM14	0570
MCM2	0561
MCM22	0567
MCM33	0563
MCM34	0552
MCM40	0746
MCM41	0565
MCM44	0564
MCM6	0555
MCT	0024
MC10	0747
MC100	0743
MC2000	0575
MC22	0560
MC3	0566
MC3034	0557
MC3336	0562
MC3543	0742
MC4000	0556
MC5000	0547
MC5074	0571
MC6774	0740
MC70	0745
MC7000	0744
MC7677	0741
MDEL	0504
MEOF	0673
MFAC	0027
MFLC1	0427
MFLD2	0435
MFLD21	0442
MFLD22	0443
MFLD3	0447
MFLD31	0454
MFLD32	0461
ML3012	0573
MMCM12	0554
MOK	0671
MSPA	0502
MTMP	0020
MX	0400
MXCHR	0465
MXLOC	0025
M2	0425
M3033	0550

M77	0551
NEXCOM	0756
NEXM	0757
NEXTM	0760
PADD	0023
PAR	0650
PLIM	0026
READ	0235
REC	0206
RETRY	0621
REVCNT	0751
REWIND	0702
SETCOM	0722
ST1	0200
ST2	0203
ST3	0216
WEOF	0246
WRDCNT	0754
WRITE	0256
XGETCHR	0031
XMAG	0323
XMX	0322

Operating instructions for the above-listed Program No. 3 are as follows:

1. Load Program (MBO).
2. Load Input tape (DRC) - Unit 1.
   Load Output tape (MX) - Unit 0.
3. SW = 200; load, clear continue.
4. Halt at 221 indicates end of input tape:
   A. To continue with same unit, replace input tape, go to instruction #3.
   B. To terminate unit, hit continue (this puts eof on output tape between units), program will then halt at 223. Go to instruction # 4-A.

Halts: 221, 223: see above.

264,647: magtape error.

652: Magtape retry failure.

Note: to get around retry failure on magtape read, SW = 207: load, continue to blank out MX record for this input. To determine whether retry is on read, check loc 752. If read, loc 752 will contain 1020.

C4. Detailed Description of Program No. 4

As previously noted, Program No. 4 constructs a complete series of mini-xm's called "10 mini-xm segments" and is accomplished only after 120 MX records are processed; at which time there are 10 sequential mini-xm words (120 bits) for each significant bit position of the 120 MX records. For ease of manipulation they are constructed partially utilizing 12 MX records at a time (12 bits in one computer word). Thus a complete group of 10 mini-xm segments consists of 456 mini-xm segments, each XM segment occupying 10 computer words (120 bits).

The manner in which the XMs are partially processed is thus:

1. Secure 12 MX records in core (one MX block).
2. Sequentially amass the first bits of each MX and store in a computer word (partial XM_1).
3. Sequentially amass the second bits of each MX and store in a computer word (partial XM_2).

4. Repeat this process for each element until all bits are processed. The resultant is a series of partial mini-xm's [xm (1,1) → xm (1,456)]

5. Repeat the above 4 steps until 10 MX blocks (120 MX records) have been thus processed. Now there are 10 words for each XM element and the mini-xm's are complete [xm (1,1) → xm (10,456)]. These mini-xm elements are stored on tape.

Program No. 4 is shown in block form at FIGURE 53 with subroutines utilized therein detailed in FIGURES 54-62. An explicit listing of the assembly level source program language for Program No. 4 and all related subroutines follows. It will be noted that the main program is shown as comprising sections I, II, III, IV, V, VI, VII and VIII in FIGURE 53. These program sections correspond to specific listing statements:

Section	Instruction Statement
I	0400-0405
II	0406-0411
III	0412-0422
IV	0423-0434
V	0435-0440
VI	0441-0462
VII	0463-0465
VIII	0466-0516

These eight program secitons are functionally described below as in introduction to the explicit program listing:

I. Initialize

A. XSETD: set disk controller and drive parameters.

B. Set up the list of data information. Eg: at the end of each input unit the program will store in this list: (1) the number of records dumped onto magtape per unit; and (2) the complement (cia) of the number of unused words from the last dump of 10 words.

C. "MONCT" is used for counting the units (indicating when the fourth is encountered).

II. Initialize small disk parameters

A. "LOOPCT" indicates that 10 input blocks of 12 records each have been processed onto the small disk, thus indicating that the small disk is full and ready to be dumped onto magtape.

B. "STAD" is an indicator for the position of the mini-xm storage on the small disk. With the first input block the first mini-xm is stored at the 10th used position of the small disk and each mini-xm of that input block is displaced 10 positions apart. With the second input block the first mini-xm is stored at the 9th used position of the small disk and again each mini-xm is displaced 10 positions apart and so forth until the disk is full.

III. Input MX Block (PIC matrix)

A. Clear the core input area. This is in effect a dummy operation since the input record overlays this area.

B. Read the MX block into core from magnetic tape. There are 12 records in this block. Each record is 64 words in length. If a tape mark is encountered, go to part VIII.

IV. Fill the small disk (Dec DF32d) with as many mini-xms as will comfortably fit (without stopping processing in the middle of an input block).

A. "XROT" - (see separate description of this subroutine) process input data into mini-xms in core mbl.

B. "XFILD" - stores the mini-xms on the disk. Note that each mini-xm word is displaced ten words apart from the next. This is done so that ten inputs of 12 records store the mini-xms in sequential order. Eg. After one input and processing of 12 MX records, the small disk contains one mini-xm for each significant MX bit position. But after 10 input and processing cycles the small disk contains 10 adjacent mini-xms for each significant MX bit position, and each "10 mini-xm segment" has its bits sequential in reverse order.

C. After "XROT" and "XFILD" are processed, if the entire unit is done or the small disk is full, part V is entered; if the small disk is not full (10 inputs), go to part III.

V. Transfer mini-xm segments from small disk to magnetic tape storage.

A. "XWRTD" is a physical transfer of the stored 10 "mini-xm segments" on the small disk to magnetic tape.

B. If the unit is completely processed go to part VI; otherwise continue to part II.

VI. End of a Unit Processing

A. Reset unit done trigger.

B. Store # of records processed in unit.

C. Store # of unused mini-xms in the last "10 mini-xm segment".

D. Write tape mark to separate units.

E. If 4 units are done, rewind the tape (of stored mini-xm segments) and go to part VII; otherwise go to part II.

VII. Transfer mini-xm segments to the large disk.

A. "XDO" - (see separate description of this subroutine.)

B. After all xms for 4 units are on the disk, rewind the magtape and halt: Program completed.

VIII. Tape mark encountered on MX input file (end of unit).

A. Set "EOFTRG" to indicate unit complete.

B. "LOOPCT" is queried to see if the last "10 mini-xm segment" has some incomplete elements stored on small disk to be dumped, in which case the last dump is made before going to part VI.

"XROT" Subroutine

To better understand how "XROT" processes the mini-xms, refer to FIGURE 26. This figure shows how each MX record is formatted in core.

Each record consists of 64 twelve bit computer words (the last word of which is not depicted on the diagram and is unused), in which there are both significant and unused bit positions. For example, the first two words consist of all significant bits, the third word, however, has only 2 significant bits and 10 unused bits. This same pattern is continued through the 48th word. The 49th, 50th, and 51st words start a new pattern: one word of 10 significant bits and 2 unused bits, then two words of all unused bits. This pattern is continued through the 60th wrod. Finally the 61st word is the last word to contain any significant bits - two bits are significant in this word.

"XROT" processes 12 records at one time. It takes a bit from each of the twelve records and puts them together and stores this result as a mini-xm. It does this for each significant MX bit position. Note that only the significant bit positions are processed into min-xms and that the number of mini-xms produced in this operation is therefore equal to the number of significant bits in the MX record.

This subroutine takes the first MX bit of each record, combines them and stores them as a mini-xm. Then it does the same with the next MX bit from each record, until the first word (12 MX bits) of each MX record is transposed into 12 mini-xm words. "MXCT" indicates when 12 records have each processed one bit. "ROTCT" indicates how many significant bits are to be processed in the current MX word.

This indicator is normally set to process 12 bits. "LETLP" indicates when the desired number of MX words have been processed to complete the 1st of the three patterns, so that the processing of only the significant bits is unaltered when the pattern changes. "LP12" indicates that the next MX word to be processed will have unused bits. It is originally set to process 2 words of each record into 24 mini-xms, then indicate done. At this time "ROTCT" is set to indicate that the next input word will have only 2 significant bits to process. "LP12" is then reset to do 3 more words (one containing significant and unused bits and the next two containing all significant bits), and so forth until the first pattern is finished, at which time 416 mini-xms will have been amassed completing the alphabetic portion of the matrix.

When the first pattern is complete "DONTRG" is set to indicate that the alpha portion is done. Notice that "ROTCT" is now set to process only 10 significant bits of the first MX word of the second pattern and the next two MX words are skipped over. This process is repeated until the second (or numeric bits) pattern is completely processed. Notice that the MX input position (data pointer) is used to test when the second pattern is done. Finally the last (or business-residence bits) pattern is processed. "ROTCT" is set to process only 2 significant bits of the next MX word. Again the input position is used to trigger the end of this pattern, at which time the subroutine is exited.

Note that each input MX word containing significant bits is processed in the forward direction (left to right) but the mini-xm bits are stored in reverse order (right to left). This is done to accommodate the first in last out function.

"XDO" Subroutine

"XDO" is actually a subprogram whose function is to combine each "ten mini-xm segment" from each magnetic tape record group into XM blocks onto a disk pack. It processes all four units processed onto tape.

First, the format of the "ten mini-xm segments" stored on magnetic tape in record groups will be described. Each of 458 bit positions of 120 MX records have been transformed into "10 mini-xm segments"; i.e., 120 bits (10 words) containing the bits of the first bit positions of each of the first 120 MX records, followed by 10 words containing the bits of the second bit position of each of the first 120 MX records, and so forth until 10 words containing the bits of the 458th bit position of each of the first 120 MX records. These are put onto magnetic tape as three records, the first and second records containing 204 (each) of the 458 segments and the third record containing the remaining 50 segments. These three records form a group. The next 120 MX records were also processed into a group of 3 records in the same manner until all records were processed (the last group possibly containing unused bits and/or words). Each of the 4 units was separated by a tape mark.

Thus "XDO" is set up to process (through "PROC" subroutine" the 4 units into XM blocks in the following manner: With the disk pack initially zeroed (offline process), the first "ten mini-xm segment" (10 mini-xm words) are put onto the first disk block (record), the second "ten mini-xm segment"put onto the second block, and so forth until the 458th "ten mini-xm segment" has been put onto the 458th disk block. Note that because the bits are in reverse order, the "ten mini-xm segments"will be put into these blocks in reverse order also, i.e., all of these segments are put into the end of the disk block (words 2509-2518). Thus the first group has been processed. The second group is processed in the same manner except that the segments are now displaced 10 words in the XM blocks (words 2499-2508). This is continued until all the groups in the unit have been processed. The other three units are processed likewise, except that the disk block are offset 1000 positions for each unit.

Some of the subroutines not shown in the FIGURES are summarized below:

MAG: General subroutine to drive the magnetic tape transport.

Format: JMS I XMAG
(command) *1
(current address)
(word count)
(extension register) *2
control resumes here if tape mark encountered
control resumes here normally

See *1 and *2 on next page

*1. command bits

0: formatter select

1,2: unit select

3-5: N/A

6-8: tape command

9-11: N/A

*2.

extension bits

0-4: N/A

5: special core dump mode

6-9: N/A

10,11: memory bank

DISKS: General subroutine to execute a disk function on the DF32D (small) disk.

Format: JMS I XDISK
(command)
(core address)
(word count)
(block # [0-17])
(disk address)
control resumes here

*3. This command is either DMAR to read or DMAW to write.

DIVIDE: General single precision divide subroutine.

Format: JMS I XDIVID
(dividend)

(divisor)

remainder returned here

control resumes here with quotient in accumulator

REWIND: General subroutine for rewinding the magnetic tape unit last processed by XMAG.

CLEAR: General subroutine to clear MX input area.

INIT: Initialize parameters at the initial read of the 3 record group of "10 mini-xm segments".

A. "RDCT" is a counter to indicate that there are 3 magtape records to one block of mini-xm segments.

B. "TRCCT" is a counter to indicate that there are 204 mini-xm segments in the first two records of the block and 50 mini-xm segments in the last record of the block. (Set to 50 at some other part of the program)

DSKOP: Finishes executing a disk function. The function is specified by "COMM". In this subroutine the actual disk transfer takes place.

RD: Establishes the parameters for "XMAG" subroutine; goes there to perform a magnetic tape read function per parameters; returns from "XMAG" indicating tape mark if encountered, and offsets return from "RD" if tape mark not encountered.

SETDSK: Sets the constant parameters for the large disk. The parameters in this program which are constant throughout the program are: controller and drive (all 4 units are processed onto one drive). "LST3" is set up at this time also. It is a list of the starting XM disk block No.'s for each unit.

Note: Each XM is a self-contained disk record, 2518 words in length. * When we speak of a disk block (on the large disk), we are denoting one of these records. Note also that there are 4060 of these blocks per disk, with the first 60 blocks used for scratch purposes, followed by 1000 blocks allotted to each unit. The breakdown of the 1000 blocks per unit is: 500 for data, 458 for XM's, 42 for scratch purposes (actually unused). For a pictorial view see the "ascii-xm file disk layout."

* Only 2500 words of the XM are used - this allows 30,000 bits per unit in the XM's.

Program No. 4.

```
            CSR=6512
            MAR=6514
            WCR=6516
            CBRL=6500
            CBRH=6501
            MCR=6517
            ESR0=6511
            AIRH=6505
            SSRH=6503
                  MQA=7501
                  MQL=7421
                  BSW=7002
                  CDF0=6201
                  CDF1=6211
                  FIXTAB
                  *36
0036  7774  CM4,   -4
0037  7775  CM3,   -3

                  *40
0040  0000  STAD,  0              /STARTING ADDRESS ON SMALL DISK
0041  0000  RECCT, 0              /# RECORDS PER MONTH
0042  0000  INBUF, 0
0043  0000  LOTRG, 0        /1-LOWER CASE ALPHA
0044  7775  RDCT,  -3       /NO READS PER DUMP
0045  7464  TRCCT, -314     /NO CHAR PER READ
0046  7720  TIMCT, -60      /NO CHAR POS PER ALPHA
0047  0000  TYPCT, 0        /NO CHARS DONE (1-62)
0050  7772  MONCT, -6       /NO MONTHS TO DO
0051  0000  OUTBUF, 0
0052  0000  DDAD,  0        /DISK ST ADDR
0053  0000  LST,   0
0054  0000  LST2,  0
0055  0000  DRIVE, 0
0056  0000  SEC,   0
```

```
0057  0012  C12,    12
0060  1400  C1400,  1400
0061  7772  CM6,    -6
0062  7766  CM12,   -12
0063  7761  CM17,   -17
0064  7464  CM314,  -314
0065  7720  CM60,   -60
0066  7776  CM2,    -2

0067  1170  XCLEAR, CLEAR
0070  0200  XMAG,   MAG
0071  0600  XDISK,  DISKS
0072  1000  XROT,   ROT
0073  0700  XFILD,  FILDSK
0074  1130  XWRTD,  WRTDSK
0075  0302  XREWD,  REWIND
0076  0735  XDO,    DO
0077  1700  XDIVID, DIVIDE
0100  0535  XINIT,  INIT
0101  0560  XRD,    RD
0102  1400  XDSK,   DSK
0103  0474  XLST,   LSTR
0104  0107  XLST2,  LSTR2
0105  1200  XPROC,  PROC
0106  1630  XSETD,  SETDSK


0107  0000  LSTR2,  0        /STORAGE OF END LOC OF MONTHS
0110  0000        0
0111  0000        0
0112  0000        0
0113  0000        0
0114  0000        0
0115  0000  FACTO,  0
                  *117
0117  0000  LST3,   0
0120  1060        1060     /START BLOCKS ON DISK
0121  3030        3030
0122  5000        5000
0123  6750        6750
0124  0000        0
                  *400              /MAINLINE
0400  7300        CLA CLL
0401  4506        JMS I XSETD       /SET CONSTANT DISK VALUES
0402  1103        TAD XLST
0403  3053        DCA LST           /STORE VARIABLES FOR EACH UNIT
0404  1036        TAD CM4
0405  3050        DCA MONCT                 /PROCESS 4 UNITS
0406  1062  SETIT, TAD CM12
0407  3270        DCA LOOPCT                /DUMP DISK EVERY 10 LOOPS
0410  1310        TAD C11
0411  3040        DCA STAD          /START FILLING DISK AT 10TH WORD
0412  4467  NXT2,  JMS I XCLEAR     /CLEAR CORE INPUT AREA
0413  1271  NXT1,  TAD C2000
0414  3217        DCA ADDR          /READ INTO 2000, MB0
0415  4470        JMS I XMAG        /READ A RECORD
0416  1020        1020
0417  0000  ADDR,  0                 /CORE ADDR
0420  6000        -2000
```

```
0421  0100          100             /EXT REG - SCD MODE
0422  5266          JMP ENDO        /EOF RETURN
0423  4472  GOON,    JMS I XROT      /YES, CHANGE TO XM
0424  4473          JMS I XFILD     /WRITE ONTO SMALL DISK
0425  7240          CLA CMA
0426  1040          TAD STAD
0427  3040          DCA STAD        /MOVE DISK LOC BACK 1
0430  1272          TAD EOFTRG
0431  7640          SZA CLA         /DONE WITH UNIT?
0432  5235          JMP DUMP        /YES
0433  2270          ISZ LOOPCT      /NO, DONE 10 TIMES - SMALL DSK FULL?
0434  5212          JMP NXT2        /NO
0435  4474  DUMP,    JMS I XWRTD     /YES, DUMP ONTO MAGTAP
0436  1272          TAD EOFTRG
0437  7650          SNA CLA         /UNIT DONE?
0440  5206          JMP SETIT       /NO
0441  3272  FINI,    DCA EOFTRG      /YES, CLEAR TRG
0442  1041          TAD RECCT
0443  3453          DCA I LST       /SAVE # RECORDS PER UNIT
0444  3041          DCA RECCT
0445  2053          ISZ LST
0446  1270          TAD LOOPCT
0447  3453          DCA I LST
0450  2053          ISZ LST         /SAVE # WORDS NEEDED FROM LAST DUMP
0451  4470          JMS I XMAG
0452  0050          50              /WRITE END OF FILE
0453  0000          0
0454  0000          0
0455  0000          0
0456  7300          CLA CLL
0457  7300          CLA CLL
0460  2050          ISZ MONCT       /4 UNITS DONE?
0461  5206          JMP SETIT       /NO
0462  4475          JMS I XREWD     /YES, REWIND TAPE
0463  4476          JMS I XDO       /PUT ONTO LARGE DISK
0464  4475          JMS I XREWD
0465  7402          HLT             /PROCESSING COMPLETE
0466  2272  ENDO,    ISZ EOFTRG      /SET DONE TRIGGER
0467  5311          JMP CKDSK       /NO
0470  0000  LOOPCT,  0
0471  2000  C2000,   2000
0472  0000  EOFTRG,  0
0473  0003  C3,      3
0474  0000  LSTR,    0               /# MAG RECORDS FOR MONTH 1
0475  0000          0               /# WORDS FILLED IN LAST RECORD
0476  0000          0
0477  0000          0
0500  0000          0
0501  0000          0
0502  0000          0
0503  0000          0
0504  0000          0
0505  0000          0
0506  0000          0
0507  0000          0
0510  0011  C11,     11
0511  1270  CKDSK,   TAD LOOPCT
0512  1057          TAD C12
0513  7640          SZA CLA         /ANY WORDS WRITTEN ON DISK?
0514  5235          JMP DUMP        /YES
0515  3270          DCA LOOPCT      /NO, DO FINAL DUMP
0516  5241          JMP FINI
                    *700
              /WRITES 1 OUTPUT BUFFER WORD BY WORD ONTO SMALL DISK;
              /204 CHARS PER TRACK
```

```
0700  0000  FILDSK,  0
0701  7300          CLA CLL
0702  3314          DCA BLOCK
0703  3312          DCA CORAD
0704  1040  RESET,   TAD STAD
0705  3315          DCA DSKAD       /ST AT LOC 12 OF DISK
0706  1064          TAD CM314
0707  3334          DCA CHARCT      /204 CHARS PER TRACK
0710  4471  WRT,     JMS I XDISK
0711  6605          DMAW            /COMMAND
0712  0000  CORAD,   0               /CORE ADDRESS
0713  7777          7777            /WORD COUNT=1
0714  0000  BLOCK,   0
0715  0000  DSKAD,   0
0716  2312          ISZ CORAD       /INC TO NEXT WORD IN CORE
0717  1312          TAD CORAD
0720  1333          TAD CM712
0721  7650          SNA CLA         /DONE?
0722  5700          JMP I FILDSK    /YES
0723  2334          ISZ CHARCT      /NO, TRACK FULL?
0724  5327          JMP INCAD       /NO, INCREMENT DISK ADDR FOR NEXT WD
0725  2314          ISZ BLOCK       /YES, GO TO NEXT TRACK
0726  5304          JMP RESET
0727  1315  INCAD,   TAD DSKAD
0730  1057          TAD C12         /INC DISK AD 10 LOCATIONS
0731  3315          DCA DSKAD
0732  5310          JMP WRT
0733  7066  CM712,   -712     /CM # WORDS FROM CORE = XM BITS
0734  0000  CHARCT,  0
                    *1000
              /CONVERTS 12 MX FRAMES INTO XM WORDS

1000  0000  ROT,     0
1001  7300          CLA CLL
1002  3051          DCA OUTBUF
1003  3300          DCA DONTRG
1004  1065          TAD CM60
1005  3301          DCA LETLP       /DO 48 ALPHA WORDS
1006  1066          TAD CM2
1007  3302          DCA LP12        /DO 2 L1 BIT WDS
1010  1271          TAD CM14
1011  3273          DCA ROTCT       /DO 12 ROTATIONS PER WORD
1012  3276          DCA STORE
1013  1271          TAD CM14
1014  3272          DCA MXCT        /DO 12 MXS
1015  1277  START,   TAD C1700
1016  3042          DCA INBUF       /MB0
1017  1042  NXMX,    TAD INBUF
1020  1274          TAD FAC         /400
1021  3042          DCA INBUF       /ORIG LOC=2000
1022  1442          TAD I INBUF
1023  7104          CLL RAL         /ROT LET POS INTO LINK
1024  3442          DCA I INBUF     /SAVE REST OF WORD
1025  1276          TAD STORE
1026  7010          RAR             /ROTATE LET POS INTO WORD
1027  3276          DCA STORE       /SAVE PARTIAL WORD
1030  2272          ISZ MXCT        /12 MXS DONE?
1031  5217          JMP NXMX        /NO, DO NEXT MX
1032  1276          TAD STORE       /YES, PUT FULL WORD IN OUTBUF
1033  6211          CDF1
1034  3451          DCA I OUTBUF
1035  2051          ISZ OUTBUF
```

```
1036  6201          CDF0
1037  2273          ISZ ROTCT       /12 INPUT ROTATIONS DONE?
1040  5264          JMP XXX
1041  2042          ISZ INBUF       /YES, INC INBUF POS
1042  1300          TAD DONTRG
1043  7640          SZA CLA         /DONE?
1044  5304          JMP QRT         /YES
1045  2301          ISZ LETLP       /NO, DONE ALPHAS?
1046  5253          JMP P2          /NO
1047  2300          ISZ DONTRG      /YES, SET TRG
1050  1062          TAD CM12
1051  3273  P3,     DCA ROTCT       /DO 10 ROTATIONS
1052  5263          JMP XXX-1
1053  2302  P2,     ISZ LP12        /DONE 2 12 BIT WDS?
1054  5261          JMP P1          /NO
1055  1303          TAD CCM3        /YES, DO 3 MORE WDS
1056  3302          DCA LP12
1057  1066          TAD CM2
1060  5251          JMP P3          /DO 2 ROT ON 1ST WD
1061  1271  P1,     TAD CM14
1062  5251          JMP P3
1063  3276          DCA STORE       /NO, CLEAR STORE
1064  1271  XXX,    TAD CM14
1065  3272          DCA MXCT
1066  1042          TAD INBUF       /NO
1067  0275          AND C77         /ACTUAL NEXT POS OF MX1
1070  5215          JMP START
1071  7764  CM14,   -14
1072  0000  MXCT,   0
1073  0000  ROTCT,  0
1074  0100  FAC,    100
1075  0077  C77,    77
1076  0000  STORE,  0               /PARTIAL OUTPUT WORD
1077  1700  C1700,  1700
1100  0000  DONTRG, 0
1101  0000  LETLP,  0
1102  0000  LP12,   0
1103  7775  CCM3,   -3
1104  1042  QRT,    TAD INBUF
1105  1314          TAD CM3372
1106  7500          SMA             /DONE NUMBERS?
1107  5315          JMP CKONIT      /YES
1110  7200          CLA             /NO
1111  2042          ISZ INBUF       /NO
1112  2042          ISZ INBUF
1113  5250          JMP P3-1
1114  4406  CM3372, -3372
1115  7640  CKONIT, SZA CLA        /DONE ONLY NUMBERS?
1116  5600          JMP I ROT       /NO, DONE BOTH
1117  2042          ISZ INBUF       /YES, DO BUS/RES SPLIT
1120  2042          ISZ INBUF
1121  1066          TAD CM2
1122  5251          JMP P3
                    *1130
              /READS DISK INTO CORE TRACK BY TRACK
              /AND WRITES ONTO TAPE
```

```
1130  0000  WRTDSK, 0
1131  1037        TAD CM3
1132  3360        DCA READCT      /READ AND WRITE 3 TRACKS
1133  3340        DCA BLKK
1134  4471  REST,   JMS I XDISK
1135  6603        DMAR            /COMMAND
1136  0000        0               /CORE ADDRESS
1137  4000        -4000           /WC
1140  0000  BLKK,   0             /BLOCK #
1141  0000        0               /DISK ADDR OFFSET(0 OR 4000)
1142  4470        JMS I XMAG
1143  0040        40              /WRITE SCD MODE
1144  0000        0               /CORE ADDR
1145  4000        -4000
1146  0101        101             /EXT REG
1147  7402        HLT             /END OF FILE RETURN
1150  2041        ISZ RECCT       /NORMAL RETURN, # REC COUNT
1151  5353        JMP .+2
1152  7402        HLT             /777RECORDS?
1153  2360        ISZ READCT      /DONE 3 TRACKS?
1154  5356        JMP .+2 /NO
1155  5730        JMP I WRTDSK    /YES
1156  2340        ISZ BLKK
1157  5334        JMP REST
1160  0000  READCT, 0
            *1170
          /CLEARS CORE INPUT AREA

1170  0000  CLEAR,  0
1171  7332        7332            /2000
1172  3042        DCA INBUF
1173  3442  AG1,    DCA I INBUF
1174  2042        ISZ INBUF       /DONE?
1175  5373        JMP AG1         /NO
1176  5770        JMP I CLEAR     /YES
1177  2000  CM6000, -6000
            *535    /CLEAR TRGS, RESET CTRS
0535  0000  INIT,   0
0536  1037        TAD CM3
0537  3044        DCA RDCT        /3 READS PER DUMP
0540  1064        TAD CM314
0541  3045        DCA TRCCT       /204 CHR PER READ
0542  5735        JMP I INIT
            *560
0560  0000  RD,     0       /READ MAG TAPE
0561  4470        JMS I XMAG
0562  0020        20      /SCD ON DRIVE 0
0563  4000        4000    /CORE ADDRESS
0564  4000        -4000   /WC
0565  0100        100     /EXT REG
0566  5760        JMP I RD        /EOF RETURN
0567  2360        ISZ RD  /NORMAL RETURN
0570  5760        JMP I RD
            *735
0735  0000  DO,     0       /PUT UNITS ONTO DISK
0736  1365        TAD C4714       /2508
0737  3052        DCA DDAD        /STARTING ADDR ON DISK
0740  1103        TAD XLST
0741  3053        DCA LST
0742  1104        TAD XLST2
0743  3054        DCA LST2
0744  1036        TAD CM4
0745  3050        DCA MONCT
```

```
0746  4500  DOWK,   JMS I XINIT      /INITIALIZE CTRS AND TRGS
0747  4505          JMS I XPROC      /PUT ONE UNIT ON DISK
0750  1052          TAD DDAD
0751  7104          CLL RAL
0752  7420          SNL      /4000 OR HIGHER ADDR?
0753  5356          JMP .+3 /NO
0754  7510          SPA      /YES
0755  7402          HLT      /OVERFLOW OF BUFR
0756  7010          RAR      /NO
0757  1057          TAD C12
0760  3454          DCA I LST2       /SAVE END LOC FOR MONTH
0761  2054          ISZ LST2
0762  2050          ISZ MONCT        /DONE 4 UNITS?
0763  5366          JMP .+3          /NO
0764  5735          JMP I DO         /YES
0765  4714  C4714,   4714
0766  1365          TAD C4714
0767  3052          DCA DDAD         /RESET ADDR
0770  5346          JMP DOWK
                /PROCESS ONE MONTH ONTO LARGE DISK
                *1200
1200  0000  PROC,    0
1201  4300          JMS RRSET
1202  7300          CLA CLL
1203  4501  READIT,  JMS I XRD
1204  5273          JMP PRCC         /EOF END OF UNIT
1205  2041          ISZ RECCT
1206  1453          TAD I LST
1207  1066          TAD CM2
1210  7041          CIA
1211  1041          TAD RECCT
1212  7640          SZA CLA          /ON LAST DUMPS OF UNIT?
1213  5224          JMP ADJ          /NO
1214  2053          ISZ LST          /YES
1215  1453          TAD I LST
1216  2053          ISZ LST
1217  7041          CIA
1220  3115          DCA FACTO        /ADJUST AS TO NO WORDS NEC
1221  1052          TAD DDAD
1222  1115          TAD FACTO
1223  3052          DCA DDAD         /ADJUST FOR NEEDED WORDS
1224  7330  ADJ,    7330     /4000
1225  3042          DCA INBUF        /ST OF DATA BUFR
1226  2044          ISZ RDCT         /NORMAL RET: 3RD READ?
1227  5232          JMP .+3 /NO
1230  1264          TAD CM62         /YES, DO 50 GRPS FROM LAST TRACK
1231  3045          DCA TRCCT
1232  4502  WRTIT,  JMS I XDSK       /WRITE 10 WORDS
1233  0000  DDBLK,   0
1234  2233  FF,     ISZ DDBLK        /INC BLOCK NO.
1235  5237          JMP .+2
1236  7402          HLT      /ERROR, BLOCK OVERFLOW
1237  2045  GG,     ISZ TRCCT        /DONE W RECORD?
1240  5232          JMP WRTIT        /NO
1241  1044          TAD RDCT         /YES
1242  7650          SNA CLA /DONE W. DUMP?
1243  5247          JMP RST          /YES
1244  1064          TAD CM314        /NO
1245  3045          DCA TRCCT        /RESET CTR
1246  5203          JMP READIT
1247  4500  RST,    JMS I XINIT      /RESET ALL CTRS, ETC.
1250  1052          TAD DDAD
```

```
1251  1062            TAD CM12
1252  3052            DCA DDAD          /SET ADDR BACK 10 LOC
1253  1052            TAD DDAD
1254  7104            CLL RAL
1255  7430            SZL       /OVERFLOW?
1256  5261            JMP .+3 /YES
1257  7200            CLA       /NO
1260  5312            JMP RQ
1261  7700            SMA CLA /OVERFLOW?
1262  5257            JMP .-3 /NO
1263  7402            HLT       /YES
1264  7716  CM62,      -62       /DO 50 CHARS ON LAST DUMP
1265  7714  CM64,      -64
1266  2260  C2260,     2260
1267  5440  CM2340,    -2340
1270  6774  C6774,     6774
1271  2041  C2041,     2041
1272  2074  C2074,     2074
1273  3041  PRCC,      DCA RECCT
1274  3115            DCA FACTO
1275  5600            JMP I   PROC
1276  7762  CM16,      -16
1277  7777  CCM1,      -1
1300  0000  RRSET,     0
1301  1517            TAD I LST3
1302  2117            ISZ LST3
1303  7450            SNA               /DONE WITH ALL UNITS?
1304  7402            HLT               /YES
1305  3311            DCA SAVE          /NO, SET STARTING DISK BLOCK FOR MOR
1306  1311            TAD SAVE
1307  3233            DCA DDBLK
1310  5700            JMP I RRSET
1311  0000  SAVE,     0
1312  1311  RQ,       TAD SAVE
1313  3233            DCA DDBLK         /RESET BLOCK NUM
1314  5203            JMP READIT
                      *1400
1400  0000  DSK,      0
1401  7300            CLA CLL
1402  1600            TAD I DSK
1403  2200            ISZ DSK
1404  3206            DCA DIVR
1405  4477            JMS I XDIVID
1406  0000  DIVR,     0         /BLOCK NO
1407  0024            24        /DIVISOR - 20
1410  0000  HD,       0         /REMAINDER
1411  3342            DCA CYL /CYL IN AC
1412  7326            CLA STL RTL       /2
1413  6517            MCR
1414  7300            CLA CLL /LOAD MODE
1415  1340            TAD C1000
1416  6501            CBRH      /SELECT CYLINDER
1417  7300            CLA CLL
1420  1342            TAD CYL
1421  7106            CLL RTL
1422  7006            RTL
1423  6500            CBRL      /CYL NO IN BITS 0-7
1424  7300            CLA CLL
1425  1060            TAD C1400
1426  6501            CBRH      /SP FN
1427  7300            CLA CLL
1430  1060            TAD C1400
```

```
1431  6500          CBRL    /SEEK AND RESET HEAD
1432  7300          CLA CLL
1433  1210          TAD HD  /SET UP HEADER IMAGE
1434  7002          BSW
1435  7110          CLL RAR         /HD IN BITS 0-6
1436  1056          TAD SEC
1437  3373          DCA HEAD        /WORD 0
1440  1342          TAD CYL
1441  3374          DCA HEAD+1      /WORD 1
1442  1373          TAD HEAD
1443  1374          TAD HEAD+1
1444  1375          TAD HEAD+2
1445  7041          CIA
1446  3376          DCA HEAD+3      /WORD 3 - CHECKSUM
1447  7325          CLA STL IAC RAL /3
1450  6517          MCR     /READ MODE
1451  7300          CLA CLL
1452  6503          SSRH    /READ SEL STATUS REG
1453  7006          RTL
1454  7006          RTL
1455  7710          SPA CLA /READY?
1456  5252          JMP .-4 /NO
1457  7326          CLA STL RTL     /YES, 2
1460  6517          MCR     /LOAD MODE
1461  7300          CLA CLL
1462  6501          CBRH            /SELECT HEAD
1463  1210          TAD HD
1464  7106          CLL RTL
1465  7006          RTL
1466  6500          CBRL    /HEAD NO BITS 3-7
1467  7300          CLA CLL
1470  1336          TAD C3400
1471  3343          DCA COMM        /CONFIRM READ
1472  4303          JMS DSKOP
1473  4744          JMS I XTRANS    /UPDATE DATA BLOCK
1474  1337          TAD C5400
1475  3343          DCA COMM        /SET TO CONF. WRITE
1476  7326          CLA STL RTL     /2
1477  6517          MCR
1500  7300          CLA CLL /LOAD MODE
1501  4303          JMS DSKOP       /WRITE BACK ONT DISK
1502  5600          JMP I DSK
1503  0000  DSKOP,  0
1504  7332          7332    /-6000
1505  6516          WCR     /SET WC TO MX
1506  7300          CLA CLL
1507  1345          TAD HEADER
1510  6514          MAR     /ST OF HDR IMAGE
1511  7300          CLA CLL
1512  1343          TAD COMM
1513  6501          CBRH    /LOAD COMMAND
1514  7300          CLA CLL
1515  1056          TAD SEC
1516  7106          CLL RTL
1517  7006          RTL
1520  6500          CBRL    /SEC NO BITS 3-7
1521  7300          CLA CLL
1522  7330          7330    /4000
1523  6505          AIRH    /SET WRITE REG IF NEC
1524  1341          TAD C100        /DATA IN B/MB1
1525  6512          CSR     /GO
1526  7325          CLA STL IAC RAL /3
1527  6517          MCR     /HEAD MODE
1530  7300          CLA CLL
```

```
1521  6511          ESR0    /READ ERROR STATUS
1532  5331          JMP .-1
1533  7510          SPA     /ERRORS ON DONE?
1534  7402          HLT     /YES
1535  5703          JMP I DSKOP     /NO
1536  3400  C3400,  3400
1537  5400  C5400,  5400
1540  1000  C1000,  1000
1541  0100  C100,   100
1542  0000  CYL,    0
1543  0000  COMM,   0
1544  1600  XTRANS, TRANS
1545  1573  HEADER, HEAD
              *1573
1573  0000  HEAD,   0       /HEAD AND SECTOR
1574  0000          0       /CYLINDER
1575  3052          -4726   /CONSTANT WC -2518
1576  0000          0       /CHECK SUM
1577  0000          0       /ADDR OF DATA - 0 MB1
              *1600
1600  0000  TRANS,  0
1601  7300          CLA CLL /UPDATE DISK BLOCK IN CORE
1602  1062          TAD CM12
1603  1115          TAD FACTO
1604  3223          DCA CTR
1605  1042          TAD INBUF
1606  1115          TAD FACTO
1607  3042          DCA INBUF
1610  1052          TAD DDAD
1611  3051          DCA OUTBUF      /ST OF WDS ON DISK BLOCK IN CORE
1612  1442  NX,     TAD I INBUF     /GET WD
1613  2042          ISZ INBUF
1614  6211          CDF1
1615  3451          DCA I OUTBUF
1616  2051          ISZ OUTBUF
1617  6201          CDF0
1620  2223          ISZ CTR /DONE?
1621  5212          JMP NX  /NO
1622  5600          JMP I TRANS     /YES
1623  0000  CTR,    0
              *1630
1630  0000  SETDSK, 0       /SET CONSTANT DISK REGISTERS
1631  1250          TAD C3000
1632  6517          MCR     /SELECT CONTROLLER 0
1633  7326          CLA STL RTL     /2
1634  6517          MCR     /LOAD MODE
1635  7300          CLA CLL
1636  7333          7333    /6000
1637  6501          CBRH    /LOAD DRIVE
1640  7300          CLA CLL
1641  1055          TAD DRIVE
1642  6500          CBRL
1643  7300          CLA CLL
1644  1247          TAD XLST3
1645  3117          DCA LST3        /ST OF BLOCK LIST
1646  5630          JMP I SETDSK
1647  0120  XLST3,  120
1650  3000  C3000,  3000
              *1700
        /DIVIDE:
        /       BOUNDS OF DIVIDEND: 0-7777
        /       BOUNDS OF DIVISOR: 1-3777
        /CALL   JMS I XDIVID
        /       (DIVIDEND)
        /       (DIVISOR)
        /       REMAINDER RETURNED HERE
        /       CONTROL RESUMES HERE WITH QUOTIENT IN AC
```

```
1700  0000  DIVIDE, 0
1701  7100        CLL
1702  3355        DCA HDIV
1703  1700        TAD I DIVIDE
1704  2300        ISZ DIVIDE
1705  3356        DCA LDIV
1706  1700        TAD I DIVIDE
1707  2300        ISZ DIVIDE
1710  7041        CIA
1711  3357        DCA DIV
1712  1355        TAD HDIV
1713  7640        SZA CLA
1714  5335        JMP DV2
1715  1356        TAD LDIV
1716  1357        TAD DIV
1717  7620        SNL CLA             /DIV<DIVISOR?
1720  5347        JMP DV4             /YES
1721  7300        CLA CLL
1722  1360        TAD CM15
1723  3361        DCA DIVCT
1724  5335        JMP DV2
1725  1355  DV3,  TAD HDIV
1726  7004        RAL
1727  3355        DCA HDIV
1730  1355        TAD HDIV
1731  1357        TAD DIV
1732  7430        SZL
1733  3355        DCA HDIV
1734  7200        CLA
1735  1356  DV2,  TAD LDIV
1736  7004        RAL
1737  3356        DCA LDIV
1740  2361        ISZ DIVCT
1741  5325        JMP DV3
1742  1355        TAD HDIV
1743  3700        DCA I DIVIDE
1744  2300        ISZ DIVIDE
1745  1356        TAD LDIV
1746  5700        JMP I DIVIDE
1747  1356  DV4,  TAD LDIV            /QUOTIENT=0, REMAINDER
1750  3700        DCA I DIVIDE        /=DIVIDEND
1751  2300        ISZ DIVIDE
1752  5700        JMP I DIVIDE
1753  2300        ISZ DIVIDE
1754  5700        JMP I DIVIDE
1755  0000  HDIV,  0
1756  0000  LDIV,  0
1757  0000  DIV,   0
1760  7763  CM15,  -15
1761  0000  DIVCT, 0

              *200
        /FORMAT:        JMS I XMAG
        /               COMMAND
        /               ADDRESS
        /               WORD COUNT
        /               EXTENSION REGISTER
        /       RETURN: EOF
        /       RETURN: NORMAL
```

```
0200  0000  MAG,    0
0201  1335          TAD MCM12
0202  3347          DCA REVCNT      /SET 10 RETRIES
0203  1600          TAD I MAG
0204  2200          ISZ MAG
0205  3350          DCA COM         /GET COMMAND
0206  7240          CLA CMA
0207  1600          TAD I MAG
0210  2200          ISZ MAG
0211  3351          DCA CADDR       /GET CURRENT ADDRESS
0212  1600          TAD I MAG
0213  2200          ISZ MAG
0214  3352          DCA WRDCNT      /GET WORD COUNT
0215  1600          TAD I MAG
0216  2200          ISZ MAG
0217  3353          DCA COMEX       /GET EXT REGISTER
0220  3346          DCA MADCOM
0221  1350  RETRY,  TAD COM
0222  1346          TAD MADCOM
0223  4322          JMS SETCOM      /SET CONTROLLER FOR FUNCTION
0224  1352          TAD WRDCNT
0225  3032          DCA 32
0226  1351          TAD CADDR
0227  3033          DCA 33          /SET WORD COUNT & CURRENT ADDRESS
0230  1353          TAD COMEX
0231  6717          6717
0232  7300          CLA CLL         /SET EXT REG
0233  4330          JMS MAGOP       /PERFORM MAGTAPE FUNTION
0234  6706          6706
0235  7421          MQL             /STORE STATUS IN MQ
0236  7501          MQA
0237  0336          AND MC6774
0240  7450          SNA
0241  5271          JMP MOK         /NO ERRORS
0242  0337          AND MC7677
0243  7450          SNA
0244  5273          JMP MEOF        /EOF
0245  0340          AND MC3543
0246  7640          SZA CLA
0247  7402          HLT             /BAD REC OR OFFLINE
0250  2347  PAR,    ISZ REVCNT      /PARITY
0251  5253          JMP .+2         /RETRY
0252  7402          HLT             /RETRY FAILURE
0253  1350          TAD COM
0254  0343          AND MC70
0255  1344          TAD MCM40
0256  7650          SNA CLA         /FAILURE ON MT7
0257  1341          TAD MC100       /YES
0260  3346          DCA MADCOM
0261  1350          TAD COM
0262  0342          AND MC7000
0263  1343          TAD MC70
0264  4322          JMS SETCOM      /SET CONTROLLER FOR BACKSPACE
0265  7240          CLA CMA
0266  3032          DCA 32          /WC = RECS BACKSPACED
0267  4330          JMS MAGOP       /PERFORM BACKSPACE
0270  5221          JMP RETRY       /AND TRY AGAIN
0271  2200  MOK,    ISZ MAG
0272  5600          JMP I MAG       /NORMAL EXIT
0273  1350  MEOF,   TAD COM
0274  0343          AND MC70
0275  1344          TAD MCM40
```

```
0276  7650          SNA CLA          /EOF ON WRITE?
0277  5250          JMP PAR          /YES - PARITY PROBLEM
0300  7000          NOP
0301  5600          JMP I MAG        /EXIT EOF
0302  0000  REWIND,  0
0303  1350          TAD COM
0304  0342          AND MC7000
0305  1345          TAD MC10
0306  4322          JMS SETCOM       /SET CONTROLLER FOR REWIND
0307  6722          6722             /EXECUTE REWIND
0310  1350          TAD COM
0311  0342          AND MC7000
0312  7106          CLL RTL
0313  7006          RTL              /LOAD CONT W/ NEXT MAG OP
0314  1355          TAD NEXM
0315  3354          DCA NEXCOM
0316  1754          TAD I NEXCOM
0317  7000          NOP
0320  7300          CLA CLL
0321  5702          JMP I REWIND
0322  0000  SETCOM,  0
0323  6711          6711
0324  5323          JMP .-1
0325  6716          6716
0326  7300          CLA CLL
0327  5722          JMP I SETCOM
0330  0000  MAGOP,   0
0331  6722          6722
0332  6701          6701
0333  5332          JMP .-1
0334  5730          JMP I MAGOP
0335  7766  MCM12,   -12
0336  6774  MC6774,  6774
0337  7677  MC7677,  7677
0340  3543  MC3543,  3543
0341  0100  MC100,   100
0342  7000  MC7000,  7000
0343  0070  MC70,    70
0344  7740  MCM40,   -40
0345  0010  MC10,    10
0346  0000  MADCOM,  0
0347  0000  REVCNT,  0
0350  0000  COM,     0
0351  0000  CADDR,   0
0352  0000  WRDCNT,  0
0353  0000  COMEX,   0
0354  0000  NEXCOM,  0
0355  0356  NEXM,    NEXTM
0356  1020  NEXTM,  1020     /0
0357  0040           40      /1
0360  0040           40      /2
0361  0040           40      /3


                     *600
              /FORMAT:       JMS I XDISK
              /              COMMAND
              /              CORE    ADDRESS
              /              WORD COUNT
              /              BLOCK NUMBER (0-17)
              /              DISK ADDRESS
```

```
0600  0000  DISKS,  0
0601  1600          TAD I DISKS
0602  2200          ISZ DISKS
0603  3236          DCA COMD1         /SET COMMAND
0604  7240          CLA CMA
0605  1600          TAD I DISKS
0606  2200          ISZ DISKS
0607  3247          DCA CAD           /SET CURRENT ADDRESS-1
0610  1600          TAD I DISKS
0611  2200          ISZ DISKS
0612  3255          DCA WCD           /SET WORD COUNT
0613  1600          TAD I DISKS
0614  2200          ISZ DISKS
0615  3250          DCA DBLK          /GET BLOXK NO
0616  1600          TAD I DISKS
0617  2200          ISZ DISKS
0620  3256          DCA DAD           /SET DISK ADDRESS
0621  1251          TAD DM3
0622  3252          DCA DREVCT        /SET FOR 3 RETRIES
0623  1255  DRETRY, TAD WCD
0624  3653          DCA I D7750       /SET WORD COUNT
0625  1247          TAD CAD
0626  3654          DCA I D7751       /SET CURRENT ADDRESS
0627  1250          TAD DBLK
0630  7110          CLL RAR
0631  7002          BSW
0632  1257          TAD C10           /MEMORY BANK 1
0633  6615          DEAL              /LOAD MEM EXT REG
0634  7210          CLA RAR
0635  1256          TAD DAD
0636  0000  COMD1,  0                 /EXECURE COMMAND
0637  6622          DFSC
0640  5237          JMP .-1           /WAIT TILL DISK DONE
0641  6621          DFSE              /ANY ERRORS?
0642  5244          JMP .+2           /YES
0643  5600          JMP I DISKS       /NO
0644  2252          ISZ DREVCT
0645  5223          JMP DRETRY        /RETRY
0646  7402          HLT               /RETRY FAILURE
0647  0000  CAD,    0
0650  0000  DBLK,   0
0651  7775  DM3,    -3
0652  0000  DREVCT, 0
0653  7750  D7750,  7750
0654  7751  D7751,  7751
0655  0000  WCD,    0
0656  0000  DAD,    0
0657  0010  C10,    10

ADDR   0417
ADJ    1224
AG1    1173
BLKK   1140
BLOCK  0714
CAD    0647
CADDR  0351
CCM1   1277
CCM3   1103
CHARCT 0734
CKDSK  0511
CKONIT 1115
CLEAR  1170
CM12   0062
CM14   1071
```

```
P3      1051
ORT     1104
RD      0560
RDCT    0044
READCT  1160
READIT  1203
RECCT   0041
RESET   0704
REST    1134
RETRY   0221
REVCNT  0347
REWIND  0302
ROT     1000
ROTCT   1073
RQ      1312
RRSET   1300
RST     1247
SAVE    1311
SEC     0056
SETCOM  0322
SETDSK  1630
SETIT   0406
STAD    0040
START   1015
STORE   1076
TIMCT   0046
TRANS   1600
TRCCT   0045
TYPCT   0047
WCD     0655
WRDCNT  0352
WRT     0710
WRTDSK  1130
WRTIT   1232
XCLEAR  0067
XDISK   0071
XDIVID  0077
XDO     0076
XDSK    0102
XFILD   0073
XINIT   0100
XLST    0103
XLST2   0104
XLST3   1647
XMAG    0070
XPROC   0105
XRD     0101
XREWD   0075
XROT    0072
XSETD   0106
XTRANS  1544
XWRTD   0074
XXX     1064
```

Operating instructions for the above listed Program No. 4 are as follows:

ABSTRACT: This program takes 12 MX records and makes 1 XM word for each char position. These are written onto the small disk until 120 frames (10 words) are stored for each position. The disk is then dumped onto magtape. After all MXs are processed, the mini-XMs (120 frames each) are then input from tape and written at the proper block on the Div Disk. At the end of the program locations 474-507 contain (1) the # of records dumped onto magtape per unit and (2) the cia of the number of unused words from the last dump of 10 words. Locations 107-115 contain the end locations of each unit within the block.

1. Load NYT MX-XM program into mb0.
2. Put MX tape on unit 1 and scratch tape on Unit 0.
3. Set the following parameters in core:
   55 = drive (0 or 20)
   56 = sector (0 or 14)
4. Be sure the drive is ready and the proper disk pack is on.
5. 400 load, clear and continue.
6. Halt at 465 indicates processing complete. Note contents of locations 474-507 and 107-115.

SPECIFICATIONS

1. Input: (a) MX tape, 1600 bpi, scd mode, 2000 wds per rec., 12 MX per record, eofs between units.

(b) Input buffer - 2000 mb0.

(c) Input buffer for mini XMs - 4000 mb0.

2. Output: (a) to small disk from 0-536 mb1.

HALT LIST

Halt Location	Reason for Halt	Recovery Procedure
247	bad record/offline	check drives
252	retry failure	press continue
465	end of processing	
646	DF32 retry failure	
755,1263	Buffer overflow; over 30,000 frames	abort
1147	Eof on writing on disk	abort
1152	record # exceeded on outputing from DF32	abort
1236	Block overflow	abort
1304	unit # exceeded	abort
1534	DD14 error	abort

D. Retrieval Processing

D1. In General.

This portion of the exemplary embodiment is generally depicted in FIGURE 63. In terms of programming, it comprises Program No. 6. It is the portion of the system which searches and retrieves the information stored on disk. The user accomplishes this function by

entering his query through the CRT keyboard into Program #6, which searches the retrieval file portion of the Diva disk to find where the query matches the disk retrieval file information. The relevant data records for these matches are then retrieved from the ASCII portion of the disk and subsequently displayed to the user via the CRT screen.

D2. Detailed Description of Program No. 6

There are two types of searches available in the system via this program. The first is a selective search. In this case, the user must specify the first letter of the "finding name", and only that segment *1 of the alpha grouping specified is searched. Otherwise, the user can search all 16 groupings (all portions) by eliminating this one word field. This is called a general search. The "finding name" is defined as the first word (Field 1) of a listing in the alphabetical telephone directory. Note that the first character (and also the second character in certain cases) of the "finding name" is the one which defines the grouping (and segment thereof) to which the listing queried will belong. Note that the listings are processed in an alpha-numeric sequence thereafter excepting that the listings and captions are not totally merged (one file merely follows the other) within a segment.

The program allows the user to enter his query in fielded format. The first field searches the first word of the "finding name" and if omitted causes the program to search the entire data base rather than one segment for any matches to subsequent information entered in the query. The second field, called "subsequent words" searches the rest of the finding name plus titles. The third field, called "address" searches the designation and address information.

The functional sequence of control can be summarized as follows:

I. User enters query through CRT.

A. Fields

1. Finding name - 7 characters max., one word only.
2. Subsequent words and titles - 6 characters max., no limit on words.
3. Designation and Address - 4 characters max., no limit on words.

B. Control Characters

1. space - field separator
2. comma - word separator
3. carriage return - query terminator for business and professional listings
4. line feed - query terminator for residential and professional listings
5. semi colon - query delector
6. period - screen hits roll
7. asterisk - abort signal

*1. Segment refers to one alphabetic segment within a grouping. Note that some alphabetic segments stand alone within the grouping (eg. "G") whereas others are combined (eg. "Z", "I", "P").

II. Program No. 6 enters search and retrieval phases

A. The characters entered from CRT are then translated to pointers for specific retrieval file blocks. The first character is a pointer to the alpha group desired. *1The first retrieval file segment (upon receipt of the next alphameric character from the CRT) is then brought into core from the Diva disk. All other retrieval file segments are ANDED (boolean operation) with the result from the previous operation. This produces a resultant retrieval file.

B. This resultant of the ANDING process is an indicator for each listing of its relation to the user's query. If a bit is "off", no match is indicated. If a bit is "on", a match with the user's query is indicated. Since the bits are sequential (with relation to the listings) in all retrieval file segments, the position of the "on" bits of the resultant retrieval file are directly proportional to the position of the ASCII data records they represent. Thus, the "hits" shown on the screen are those ASCII records represented by the "on" bits in the resultant retrieval file.

For ease of reference, each retrieval file segment will hereafter be termed as an "XM".

*1. The first character received from the CRT in a search is a director to the a) proper disk pack, b) sector within a pack, and c) one of 4 segments (groups) within a sector, ie., it is a pointer to one of the 16 alpha groupings.

As an example of the general processing involved, consider the following query:

Smith John,H 4, Park (Return)

Upon receipt of each letter of this query, Program No. 6 will undertake the following actions:

S - indicate split letter segment

M - set disk parameters for alpha grouping #13 (SA-SN). bring XM block 12 into core (this becomes resultant).

I - bring XM block 34 into core; and with resultant and result becomes resultant.

T - same (XM 71)

H - same (XM 85)

J - same (XM 165)

O - same (XM 196)

H - same (XM 215)

N - same (XM 247)

H - same (XM 163)

4 - same (XM 420)

P - same (XM 327)

A - same (XM 338)

R - same (XM 381)

K - same (XM 400)

Return - same (XM 457) - business professional filter

Program No. 6 is shown in block form at FIGURES 64-68 with some of the subroutines used therein detailed in FIGURES 69-84. An explicit listing of the assembly level source program language for Program No. 6 and all related

subroutines follows. It will be noted that the main program is shown as comprising sections I - IX in FIGURES 64-68. These program sections correspond to specific listing statements:

Section	Instruction Statements
I	0200-0202
II	0203-0210
III	0211-0222
IV	0223-0226
V	0400-0473
VI	0227-0315
VII	0316-0345
VIII	1317-1354
IX	0600-0657; 0736-0767

These nine program sections are functionally described below as an introduction to the explicit program listing:

I. Program Initialization

A. XPRINT: print the form on the CRT. The form consists of dashes denoting where the query fields should be entered.

II. Start of Query.

A. "6030" clears the keyborad flag if set.

B. "FPOS" is set to indicate the program is processing the first field (FPOS = 0).

C. "CHAR" is set to indicate that no characters of the query have been processed.

D. A buffer is set to store XM coordinates for each query character if the whole disk is to be searched.

E. Note that because the field is already set, the new field processing is ignored at this time.

III. New Field

A. Only three fields are allowed, if the fourth is attempted, the program aborts the query processing and reinitializes itself.

B. "CHAR" is queried to see if any query characters have been entered. If not, the program (upon the omission of any characters in the first field) determines that the search will cover the entire file (all 16 segments). If the entire file is to be searched, a resultant XM and parameters which are pertinent there to are stored for each of the 16 segments. "XSTBUF" is an initialization routine which stores one XM (with all bits set) and its initialized parameters for each segment. For a brief explanation of these parameters; "XMBIT" is the bit which is currently being tested for a hit, "XMLOC" is the location which contains the bit currently being tested for a hit, and rather than processing the entire XM for bits during the anding and searching cycles, "XMSTRT" points to the first "on" bit of the resultant, so that processing can start at this location.

C. "FPOS" is set to indicate the new field (FPOS = 1 for second field; FPOS = 2 for the third)

IV. New Word - At the beginning of a word, certain items must be reinitialized.

A. "CPOS" - used to denote character position within the word of the current character being processed. Initially set to 0 for the first character.

B. "SADD" - set to the starting XM block # for the current character position within a field. See FIGURE 26.

C. "STRG" - When the first character of the finding name field is one which is split among more than one XM group (eg. "S"), "STRG" is set so that the next character can determine the XMs group to search.

D. An exit is made to "XASCIF" to get the next character from the CRT.

V. "XASCIF"

A. "KYBD" subroutine gets a character from the CRT keyboard and stores it in MQ register.

B. Program control is routed per MQ contents. For routing, see the flow chart on this section, and its accompanying legend.

VI. Current Character is an Upper Case Alpha

A. If "CPOS" and "FPOS" combined are equal to zero, this indicates that the current character is the first character of the finding name and

1. "GET1" selects the proper XM group and its disk parameters: drive, sector, and starting cylinder.
2. "STRG" is set if first characters denotes split group, so that proper disk parameters can be determined upon encountering the next character.
3. Parameters for the query are initialized (See new field section for description of "XMBIT", "XMLOC" and "XMSTRT").

4. "CHAR" is set to indicate that the search will include only one segment.

B. "STRG" is queried to see if disk parameters must be set on current character (when previous character could not set parameters because of segment split). And if so,

1. "STRG" is rezeroed for remainder of query.
2. "XGET2" sets the proper disk parameters with current character determining the proper XM group to be searched.

C. "LIMST" list tells how many characters are allowed to be processed in each word in a field. (Actually it is a list composed of the last valid character position of the field).

Field 1 - 7 characters

Field 2 -6 characters

Field 3 - 4 characters

If "CPOS" is greater than that allowed in the "LIMST" list for that field, character processing is ignored for the current character.

D. "FLDST" list is a list of the starting XM block numbers for each field (see FIGURE 26 for block numbers)

Field 1 - Block 0

Field 2 - Block 156 (234 octal)

Field 3 - Block 312 (470 octal)

E. The actual XM block to be processed is determined and put in "STMP". It is determined by adding the starting block of the current field (from "FLDST" list-pointed to by "STMP" current contents), the starting block within the field of the current character position (in "SADD") and the stripped character (MQ register contents stripped of its ASCII).

F. "SADD" is then reset for the next character position.

G. "CHAR" is queried to determine whether all segments are to be processed for the current character; and if so, the XM block number to be retrieved is stored in a buffer via auto-index register "R17" (disk processing done at end of query) and immediate processing is bypassed.

H. Finally the block number (stored in "STMP" is divided by 20 (decimal)) and this result is added to the starting cylinder for this segment to determine the proper disk record to process. "XDOIT" is the ANDING subroutine which processes this disk record. Method: the records are read into core and ANDED together, the resultant containing only those bits which refer to matches in the query.

I. "CPOS" is incremented to the next character position. "CHAR" information is transferred to "ECHO" for further reference.

J. Program control goes to part IV to get next character from CRT.

VII. Current Character is a Number

A. "FPOS" is queried to see if the current field is the third (FPOS = 2) - if not an error condition exists and a "no hits condition" is rendered to the user on the CRT screen.

B. Again "CPOS" is tested to see whether the character position is greater than the 4th (OPOS 3), and if so, processing is ignored.

C. The block number formulation is similar to that of the alphabetics except that the character position is multiplied by 10 to substitute for the "SADD" element and the starting block number for the field is set to 416 (640 octal) - see "MX -XM formats".

D. Remainder of processing is same as for alphabetics.

VIII. End of query

A. The last key selected in query determines type of search. Actually it allows one more ANDING process to occur (either the business - professional block or the residential - professional block noted in "BPRADD".

B. "ECHO" is tested to see if all segments are being searched. If so,

1. This last block # is stored in the XM block buffer and the buffer is terminated (0).

2. "D016" processes all XM blocks whose numbers are stored in buffer.

3. "XGET16" brings the first segment's resultant and parameters into core. "UNIT" is set to donote first segment.

4. Program control is given to "XHIT".

C. If only one sigment is being processed, the last block is ANDED into the resultant via "XDOIT" and program control is given to "XHIT".

IX. Translate "on" bits in query's resultant XM into query answers (original listings) on CRT screen.

A. "XPRINT" at "MOVCUR" moves the cursor to the answer section of the CRT screen.

B. "HITCNT" is set to allow a maximum of 18 hits to be displayed on the CRT screen.

C. The current bit being processed in the resultant is stored in the MQ register.

D. Program control is transferred to "H32" to continue scanning for "on"bits.

E. "H20" - Scan resultant (computer) words for "on" bits.

1. "XMLOC" (current resultant location being scanned) is tested for "on" bits, and if any are encountered program control is given to "H30".

2. "XMLOC" is queried to see if the scanning process has been completed (if "XMLOC" is at beginning of resultant - note scanning done backwards) and if so, program control is given to "ENDH".

3. "XMLOC" is reset back one location to scan next location; program control is then given to "H21".

F. "H30" - The "on" bits are determined and processed.

1. Going from right to left, each bit is tested and if on program control is given to "H40", which processes the "on" bit.

2. After all bits are checked, program control is returned to "H22" to continue bit scan.

G. "H40" - Proper listing which matches "on" bit is written onto CRT screen.

1. "DOHIT" - selects relative ASCII record per the "on" bit in resultant and "XPHIT" prints this record on the CRT screen.

2. "HITCNT" is queried to see if CRT screen is full (18 listings printed), and if not program control is returned to "H32" to continue processing any "on" bits left in resultant.

3. If screen is full, the current bit being processed is returned to storage in "XMBIT"; "HITCNT" is queried to see if any hits were detected in resultant and if not - "no hits indication" is presented to user on CRT screen; the cursor is returned to the start of the query on the CRT screen; program control is returned to "XST2" to process a new query or possibly more hits if selected by user.

H. "ENDH" - end of resultant processing.

1. "ECHO" is queried to see whether more than one segment is being processed, and if not - program control is given to "H41" to terminate processing.

2. After four segments' resultants have been processed it is necessary to change the disk sector and

drive parameters, otherwise "XGET16" will bring the resultant (and parameters) specified in call by "UNIT" into core. Program control is returned to "HIT14".

3. If when specifying the new drive-sector, it is determined that all segments have already been processed, program control is given to "H41" to terminate processing.

Some of the subroutines not explicitly shown in FIGURES 69-84 are briefly summarized below:

KYBD: gets character from CRT keyboard and stores it in MQ register.

PRT: prints characger currently residing in accumulator onto CRT screen.

PRINT: prints all characters in list specified in the call format.

format: JMS PRINT
(location of print characters list)
control resumes here

NOTE: list is terminated by ϕ.

DIVIDE: general purpose single precision division subroutine.

format: JMS I XDIVID
(dividend)
(divisor)
remainder returned here
control resumes here with quotient in accumulator

ZMX: extraneous subroutine which is not used in program.

DISK: general purpose subroutine which drives the disk hardware to perform an operation (specified in call).

format: JMS I XDISK

[memory bank(bits 0-5) & drive (bits 6-11)]

[cylinder]

[sector (bits 0-5) and head (bits 6-11)]

[command]

control resumes here

The CRT keyboard is utilized as follows:

UCA - upper case alpha; query character key

\# - number; query character key

LF - line feed; key to indicate end of query and specify residential-professional type search

CR - carriage return; key to indicate end of query and specify business-professional type search

SP - space; key to indicate new field

* - asterisk; key to indicate abort situation

, - comma; key to indicate end of word

. - period; key to indicate screen roll (more hits)

; - semi-colon; key to indicate deletion of query

other- any other characters entered are not penetrable into the system (thus a question mark-backspace is printed for user).

Program No. 6

```
BSW=7002
MQL=7421
MQA=7501
SWP=7521
CDF0=6201
CDF1=6211
CDF2=6221
DMCR=6517
DCBRH=6501
DCBRL=6500
DSSRH=6503
DSSRL=6502
DESR0=6511
DESR1=6515
DWCR=6516
DDMAR=6514
DCSR=6512
DAIRL=6504
```

```
            DAIRH=6505
            DUSRL=6506
            DUSRH=6507
            GTF=6004
            RTF=6005
            RMF=6244
            RIB=6234
            SRQ=6003

            FIXTAB

            /ADDRESS TABLE: XM DEMO 2

                        *7
0007  0000  DRIVER, 0

            /DISSLY PROGRAM LABEL
                        *0
                        304;
0000  0304
0001  0311  311;
0002  0323  323;
0003  0323  323;
0004  0314  314;
0005  0331  331;
0006  0330  330;
0007  0315  315

                        *10
0010  0000  R10,    0
0011  0000  R11,    0
0012  0000  R12,    0
0013  0000  R13,    0
0014  0000  R14,    0
0015  0000  R15,    0
0016  0000  R16,    0
0017  0000  R17,    0

0020  3052  WC,     3052    /WORD COUNT
0021  3052  CA,     -4726   /CURRENT ADDRESS (END OF BUFFER)
0022  0000  XMLOC,  0       /CURRENT HIT LOC
0023  0000  XMBIT,  0       /CURRENT HIT BIT
0024  0000  XMSTRT, 0       /START OF XM BUFFER
0025  0000  XCYL,   0       /STARTING CYLINDER OF GROUP

0026  0000  HITCNT, 0
0027  0000  DSEC,   0
0030  0000  UNIT,   0
0031  0000  CHAR,   0
0032  0000  ECHO,   0
0033  0000  BPRADD, 0

0034  0600  XHIT,   HIT
0035  2341  XGET16, GET16
0036  2314  XPUT16, PUT16
0037  0510  XPRINT, PRINT
0040  0474  XKYBD,  KYBD
0041  1200  XDIVID, DIVIDE
0042  1415  XDISK,  DISK
0043  1275  XDOIT,  DOIT
0044  0203  XST2,   QRY
0045  2000  XPHIT,  PHIT
0046  0343  XNONE,  NERR
```

```
0047  0710  XC234,  710
0050  3074  XC3074, 3074
0051  1400  XC1400, 1400
0052  7774  XCM4,   -4
0053  0062  XC62,   62
0054  0034  X34,    34
0055  3052  L3052,  3052
0056  3053  L3053,  3053
0057  3054  L3054,  3054
0060  3055  L3055,  3055
0061  3000  STBUF,  CHARBF

      PAGE

      /MAINLINE:      NYT RETRIEVAL

0200  7300  ST1,    CLA CLL /INITIALIZE
0201  4437          JMS I XPRINT
0202  2600          FORM

0203  6030  QRY,    6030    /NEW QUERY
0204  3355          DCA FPOS
0205  3031          DCA CHAR
0206  1061          TAD STBUF
0207  3017          DCA R17
0210  5223          JMP WRD

0211  1355  FLD,    TAD FPOS        /NEW FIELD
0212  1352          TAD SCM2
0213  7650          SNA CLA /FIELD LIM OVERRUN?
0214  5200          JMP ST1         /YES, ABORT SEARCH IF TOO MANY FIELD
0215  1355          TAD FPOS
0216  1031          TAD CHAR
0217  7650          SNA CLA
0220  4760          JMS I XSTBUF    /1ST FLD OMITTED - SEARCH ALL
0221  2355  FLDRTN, ISZ FPOS
0222  7000          NOP

0223  3354  WRD,    DCA CPOS        /NEW WORD
0224  3356          DCA SADD
0225  3357          DCA STRG

0226  5761  SNX1,   JMP I XASCIF   /NEW CHAR - FILTER

      /RETURNS FROM ASCIF

0227  1354  UCA,    TAD CPOS        /UPPER CASE CHAR
0230  1355          TAD FPOS
0231  7650          SNA CLA
0232  5303          JMP SFST        /1ST CHR OF FINDING FIELD
0233  1357          TAD STRG
0234  7640          SZA CLA
0235  5300          JMP SSEC        /2ND CHR OF FINDING FLD ON CHR SPLIT
0236  1355  SNXC,   TAD FPOS
0237  1364          TAD LIMST
0240  3264          DCA STMP
0241  1664          TAD I STMP
0242  1354          TAD CPOS
0243  7740          SMA SZA CLA
0244  5271          JMP SNXT2       /FIELD LIMIT OVERFLOW
0245  1355          TAD FPOS
```

```
0246  1370          TAD FLDST
0247  3264          DCA STMP
0250  7501          MQA     /BLK=STRIP CHR X POS + FLD START
0251  1353          TAD SCM301
0252  1356          TAD SADD
0253  1664          TAD I STMP
0254  3264  NXU,     DCA STMP
0255  1356          TAD SADD
0256  1350          TAD SC32
0257  3356          DCA SADD
0260  1031          TAD CHAR
0261  7650          SNA CLA
0262  5275          JMP PR16        /1ST FLD OMITTED - PROCESS ALL
0263  4441          JMS I XDIVID
0264  0000  STMP,    0
0265  0024          24
0266  0000          0       /REMAINDER:     HEAD
0267  1025          TAD XCYL        /CYL + CYL STRT
0270  4443          JMS I XDOIT     /ANDING PROCESS
0271  2354  SNXT2,   ISZ CPOS
0272  1031          TAD CHAR
0273  3032          DCA ECHO
0274  5226          JMP SNX1

0275  1264  PR16,    TAD STMP
0276  3417          DCA I R17
0277  5271          JMP SNXT2

0300  3357  SSEC,    DCA STRG
0301  4763          JMS I XGET2     /GET DISK PRMS & CHR XM BUF
0302  5236          JMP SNXC

0303  4762  SFST,    JMS I XGET1     /GET DISK PRMS & CHR XM BUF
0304  2357          ISZ STRG        /RETURN IF B,C,M,S (SPLIT CHRS)
0305  7330          CLA CLL CML RAR
0306  3023          DCA XMBIT
0307  3022          DCA XMLOC
0310  1351          TAD SC3074
0311  3024          DCA XMSTRT
0312  4437          JMS I XPRINT
0313  2717          LET1
0314  2031          ISZ CHAR        /INDICATE 1ST FLD PRESENT
0315  5271          JMP SNXT2

0316  1355  NUM,     TAD FPOS        /NUMBER RETURN
0317  1352          TAD SCM2
0320  7640          SZA CLA /ADDRESS FIELD?
0321  5343          JMP NERR        /NO - ERROR
0322  1354          TAD CPOS
0323  1367          TAD LIMST+3
0324  7740          SMA SZA CLA
0325  5271          JMP SNXT2       /FIELD LIM OVERFLOW
0326  1354          TAD CPOS
0327  7106          CLL RTL
0330  7004          RAL
0331  3264          DCA STMP
0332  1354          TAD CPOS
0333  7104          CLL RAL
0334  1264          TAD STMP
0335  1347          TAD SC640
```

```
0336  3264          DCA STMP
0337  7501          MQA
0340  1346          TAD SCM260
0341  1264          TAD STMP
0342  5254          JMP NXU

0343  4437  NERR,   JMS I XPRINT
0344  2701          NOHITS
0345  5203          JMP QRY

0346  7520  SCM260, -260
0347  0640  SC640,  640
0350  0032  SC32,   32
0351  3074  SC3074, 3074
0352  7776  SCM2,   -2
0353  7477  SCM301, -301

0354  0000  CPOS,   0
0355  0000  FPOS,   0
0356  0000  SADD,   0
0357  0000  STRG,   0

0360  2200  XSTBUF, SETBUF
0361  0400  XASCIF, ASCIF
0362  1000  XGET1,  GET1
0363  1027  XGET2,  GET2

0364  0365  LIMST,  .+1
0365  7772          -6
0366  7773          -5
0367  7775          -3
0370  0371  FLDST,  .+1
0371  0000          0
0372  0234          234
0373  0470          470

          PAGE

0400  4274  ASCIF,  JMS KYBD        /GET CHAR
0401  7501          MQA
0402  1325          TAD ACM260
0403  7510          SPA
0404  5216          JMP CKR
0405  1326          TAD ACM12
0406  7510          SPA
0407  5266          JMP ANUM        /260<=X<=271:   #
0410  1327          TAD ACM7
0411  7510          SPA
0412  5216          JMP CKR
0413  1330          TAD ACM32
0414  7510          SPA
0415  5271          JMP AUCA        /301<=X<=332:   UCA

0416  7701  CKR,    CLA MQA
0417  1331          TAD ACM212
0420  7450          SNA
0421  5744          JMP I XRP       /212:   LF (RES-PROF SEARCH)
0422  1332          TAD ACM3
0423  7450          SNA
0424  5745          JMP I XBP       /215:   CR (BUS-PROF SEARCH)
0425  1333          TAD ACM23
```

```
0426  7450          SNA
0427  5251          JMP ATAB        /240:   SP (END OF FIELD)
0430  1326          TAD ACM12
0431  7450          SNA
0432  5754          JMP I XDMON     /252:   * (RETURN TO MONITOR)
0433  1334          TAD ACM2
0434  7450          SNA
0435  5254          JMP ACOM        /254:   , (END OF WORD)
0436  1334          TAD ACM2
0437  7450          SNA
0440  5434          JMP I XHIT      /256:   . (ROLL SCREEN)
0441  1335          TAD ACM15
0442  7650          SNA CLA
0443  5257          JMP ADEL        /273:   ; (DELETE QUERY)
0444  1336          TAD AC277
0445  4302          JMS PRT /FOR OTHER CHRS PRINT:  ? BKSP
0446  1337          TAD AC210
0447  4302          JMS PRT
0450  5200          JMP ASCIF

0451  1340  ATAB,   TAD AC211
0452  4302          JMS. PRT
0453  5746          JMP I XFLD

0454  7701  ACOM,   CLA MQA
0455  4302          JMS PRT
0456  5750          JMP I XWRD

0457  1341  ADEL,   TAD AC215
0460  4302          JMS PRT
0461  1342          TAD AC243
0462  4302          JMS PRT
0463  1343          TAD AC253
0464  4302          JMS PRT
0465  5751          JMP I XQRY

0466  7701  ANUM,   CLA MQA
0467  4302          JMS PRT
0470  5753          JMP I XNUM

0471  7701  AUCA,   CLA MQA
0472  4302          JMS PRT
0473  5752          JMP I XUCA

0474  0000  KYBD,   0
0475  6331          6331
0476  5275          JMP .-1
0477  6336          6336
0500  7421          MQL
0501  5674          JMP I KYBD

0502  0000  PRT,    0
0503  6346          6346
0504  6341          6341
0505  5304          JMP .-1
0506  7300          CLA CLL
0507  5702          JMP I PRT

0510  0000  PRINT,  0
0511  7240          CLA CMA
0512  1710          TAD I PRINT
0513  2310          ISZ PRINT
0514  3011          DCA R11
0515  1411  NXZPR,  TAD I R11
```

```
0516  7450          SNA                 /END?
0517  5710          JMP I PRINT         /YES
0520  6346          6346
0521  6341          6341
0522  5321          JMP .-1
0523  7300          CLA CLL
0524  5315          JMP NXZPR


0525  7520  ACM260, -260
0526  7766  ACM12,  -12
0527  7771  ACM7,   -7
0530  7746  ACM32,  -32
0531  7566  ACM212, -212
0532  7775  ACM3,   -3
0533  7755  ACM23,  -23
0534  7776  ACM2,   -2
0535  7763  ACM15,  -15
0536  0277  AC277,  277
0537  0210  AC210,  210
0540  0211  AC211,  211
0541  0215  AC215,  215
0542  0243  AC243,  243
0543  0253  AC253,  253

0544  1320  XRP,    RP
0545  1317  XBP,    BP
0546  0211  XFLD,   FLD
0547  0200  XST1,   ST1
0550  0223  XWRD,   WRD
0551  0203  XQRY,   QRY
0552  0227  XUCA,   UCA
0553  0316  XNUM,   NUM
0554  0200  XDMON,  200

          PAGE


0600  4437  HIT,    JMS I XPRINT
0601  2674          MOVCUR              /MOVE CURSOR TO HITS
0602  1373          TAD HCM22           /SET UP FOR 18 HITS
0603  3026          DCA HITCNT
0604  1023          TAD XMBIT
0605  7421          MQL
0606  5233          JMP H32

0607  6211  H20,    CDF1    /FIND HIT
0610  1422  H21,    TAD I XMLOC
0611  7440          SZA     /HIT?
0612  5224          JMP H30 /YES
0613  1022  H22,    TAD XMLOC
0614  7041          CIA
0615  1024          TAD XMSTRT
0616  7650          SNA CLA /DONE?
0617  5336          JMP ENDH            /YES
0620  7240          CLA CMA
0621  1022          TAD XMLOC
0622  3022          DCA XMLOC
0623  5210          JMP H21
```

```
0624  3374  H30,    DCA HHIT          /FIND HIT BIT
0625  7324          CLA CLL CML RAL /SET BIT 11
0626  7421  H31,    MQL
0627  7501          MQA
0630  0374          AND HHIT
0631  7640          SZA CLA /THIS BIT?
0632  5241          JMP H40 /YES
0633  7501  H32,    MQA
0634  7104          CLL RAL
0635  7420          SNL     /ALL BITS DONE?
0636  5226          JMP H31 /NO
0637  6211          CDF1
0640  5213          JMP H22

0641  6201  H40,    CDF0    /PROCESS HIT
0642  4260          JMS DOHIT         /PER XMLOC & MQ BIT
0643  2026          ISZ HITCNT        /ALL HITS DONE?
0644  5233          JMP H32 /NO
0645  6201  H41,    CDF0
0646  7501          MQA
0647  3023          DCA XMBIT
0650  1026          TAD HITCNT
0651  7041          CIA
0652  1373          TAD HCM22
0653  7650          SNA CLA
0654  5446          JMP I XNONE       /NO HITS
0655  4437          JMS I XPRINT
0656  2715          CURHOM
0657  5444          JMP I XST2

0660  0000  DOHIT,  0
0661  1022          TAD XMLOC
0662  1372          TAD HM3074
0663  3265          DCA HTMP
0664  4441          JMS I XDIVID      / POS/5 = BLK + X
0665  0000  HTMP,   0
0666  0005          5
0667  0000  HX,     0       /REMAINDER
0670  3272          DCA HBLK
0671  4441          JMS I XDIVID      / BLK/20 = CYL + HED
0672  0000  HBLK,   0
0673  0024          24
0674  0000          0       /REMAINDER
0675  1025          TAD XCYL          /CYL + CYL STRT
0676  1371          TAD HCM31
0677  3305          DCA HCYL
0700  1274          TAD HBLK+2
0701  1027          TAD DSEC
0702  3306          DCA HHED
0703  4442          JMS I XDISK       /INPUT DATA INTO MB2
0704  0200          200
0705  0000  HCYL,   0
0706  0000  HHED,   0
0707  3400          3400
0710  3265          DCA HTMP
0711  7501          MQA     /PUT BIT IN DIGITAL FORMAT
0712  7110          CLL RAR
0713  7430          SZL
0714  5317          JMP .+3
0715  2265          ISZ HTMP
0716  5312          JMP .-4
0717  7300          CLA CLL
0720  1265          TAD HTMP
```

```
0721  1370          TAD HCM13
0722  7041          CIA
0723  3265          DCA HTMP
0724  1267          TAD HX  / 12X + BIT = REC
0725  7106          CLL RTL
0726  7004          RAL
0727  3272          DCA HBLK
0730  1267          TAD HX
0731  7106          CLL RTL
0732  1272          TAD HBLK
0733  1265          TAD HTMP
0734  4445          JMS I XPHIT       /PRINT HIT (REC # IN AC)
0735  5660          JMP I DOHIT

0736  6201  ENDH,    CDF0
0737  1032          TAD ECHO
0740  7640          SZA CLA           /MORE THAN ONE SEGMENT?
0741  5245          JMP H41           /NO
0742  1030          TAD UNIT
0743  1052          TAD XCM4
0744  7650          SNA CLA           /DONE 4 UNITS?
0745  5354          JMP ENDH20        /YES - CHANGE DRIVE-SECTOR IF MORE
0746  2030  ENDH10, ISZ UNIT
0747  1030          TAD UNIT
0750  3352          DCA .+2
0751  4435          JMS I XGET16      /GET NEXT RES; PRMS
0752  0000          0
0753  5204          JMP HIT+4
0754  1027  ENDH20, TAD DSEC
0755  1007          TAD DRIVER
0756  7450          SNA
0757  5245          JMP H41           /ALL 16 SEGMENTS DONE
0760  7110          CLL RAR           /RESET SECTOR-DRIVE:
0761  7630          SZL CLA           /1400-1 -->     1400-0
0762  5365          JMP ENDH30        /1400-0 -->     0-1
0763  3027          DCA DSEC          /0-1    -->     0-0
0764  7201          CLA IAC
0765  3007  ENDH30, DCA DRIVER
0766  3030          DCA UNIT
0767  5346          JMP ENDH10

0770  7765  HCM13,  -13
0771  7747  HCM31,  -31
0772  4704  HM3074, -3074
0773  7756  HCM22,  -22
0774  0000  HHIT,   0

              PAGE


1000  0000  GET1,   0
1001  3327          DCA GCT
1002  7501          MQA
1003  1336          TAD GCM302
1004  7450          SNA
1005  5222          JMP G11 /B
1006  2327          ISZ GCT
1007  1337          TAD GCM1
1010  7450          SNA
1011  5222          JMP G11 /C
1012  2327          ISZ GCT
1013  1340          TAD GCM12
1014  7450          SNA
1015  5222          JMP G11 /M
```

```
1016  2327          ISZ GCT
1017  1341          TAD GCM6
1020  7640          SZA CLA
1021  2200          ISZ GET1          /REG CHR

1022  7501  G11,     MQA
1023  1342          TAD GCM301
1024  3251          DCA GCHR
1025  4247          JMS GSET
1026  5600          JMP I GET1

1027  0000  GET2,    0
1030  1327          TAD GCT
1031  1350          TAD GTAB1
1032  3347          DCA GSAVE
1033  7501          MQA
1034  1747          TAD I GSAVE
1035  7700          SMA CLA
1036  5242          JMP GRET2         /2ND CHR AFTER BREAK
1037  5627          JMP I GET2        /2ND CHR BEFORE BREAK
1040  7000          NOP
1041  7000          NOP
1042  1344  GRET2,  TAD GC32
1043  1327          TAD GCT
1044  3251  GRET3,  DCA GCHR
1045  4247          JMS GSET
1046  5627          JMP I GET2

1047  0000  GSET,    0
1050  4270          JMS ZXM           /SET UP XM BUF
1051  0000  GCHR,    0
1052  1251          TAD GCHR
1053  1346          TAD GAONE
1054  3251          DCA GCHR
1055  1651          TAD I GCHR        /4XXX:  DRIVE
1056  7710          SPA CLA
1057  7201          CLA IAC
1060  3007          DCA DRIVER
1061  1651          TAD I GCHR        /14XX:  SECTOR
1062  0343          AND GC1400
1063  3027          DCA DSEC
1064  1651          TAD I GCHR        /X377:  STARTING CYLINDER
1065  0345          AND GC377
1066  3025          DCA XCYL
1067  5647          JMP I GSET

              /ZXM,   ZEROES ALL IRRELEVANT AREAS OF XM
              /       SETS REST TO 1'S
              /CALL:  JMS I XZM
              /       CHAR (0-35)

1070  0000  ZXM,     0
1071  1670          TAD I ZXM
1072  1330          TAD LBND
1073  3332          DCA ZLLIM
1074  1670          TAD I ZXM
1075  1331          TAD UBND
1076  3333          DCA ZULIM
1077  2270          ISZ ZXM
1100  1732          TAD I ZLLIM
```

```
1101  3332           DCA ZLLIM
1102  1733           TAD I ZULIM
1103  3333           DCA ZULIM
1104  1334           TAD ZC3074
1105  3335           DCA ZX
1106  6211  ZNX1,    CDF1
1107  3735           DCA I ZX
1110  6201           CDF0
1111  2335           ISZ ZX
1112  5306           JMP ZNX1
1113  5315           JMP .+2
1114  2332  ZNX2,    ISZ ZLLIM
1115  7240           CLA CMA
1116  6211           CDF1
1117  3732           DCA I ZLLIM
1120  6201           CDF0
1121  1332           TAD ZLLIM
1122  7041           CIA
1123  1333           TAD ZULIM
1124  7640           SZA CLA
1125  5314           JMP ZNX2
1126  5670           JMP I ZXM

1127  0000  GCT,     0
1130  2436  LBND,    LBOUND
1131  2474  UBND,    UBOUND
1132  0000  ZLLIM,   0
1133  0000  ZULIM,   0
1134  3074  ZC3074,  3074
1135  0000  ZX,      0
1136  7476  GCM302,  -302
1137  7777  GCM1,    -1
1140  7766  GCM12,   -12
1141  7772  GCM6,    -6
1142  7477  GCM301,  -301
1143  1400  GC1400,  1400
1144  0032  GC32,    32
1145  0377  GC377,   377
1146  2400  GAONE,   AONE
1147  0000  GSAVE,   0

1150  1151  GTAB1,   .+1
1151  7456           -322
1152  7456           -322
1153  7467           -311
1154  7461           -317

      PAGE


      /XDIVID SINGLE PRECISION DIVIDE SUBROUTINE
      /       BOUNDS OF DIVIDEND: 0-7777
      /       BOUNDS OF DIVISOR: 1-3777
      /CALL   JMS I XDIVID
      /       (DIVIDEND)
      /       (DIVISOR)
      /       REMAINDER RETURNED HERE
      /       CONTROL RESUMES HERE WITH QUOTIENT IN AC
```

```
1200  0000  DIVIDE, 0
1201  7300        CLA CLL
1202  3253        DCA HDIV
1203  1600        TAD I DIVIDE
1204  2200        ISZ DIVIDE
1205  3254        DCA LDIV
1206  1600        TAD I DIVIDE
1207  2200        ISZ DIVIDE
1210  7041        CIA
1211  3255        DCA DIV
1212  1253        TAD HDIV
1213  7640        SZA CLA
1214  5235        JMP DV2
1215  1254        TAD LDIV
1216  1255        TAD DIV
1217  7620        SNL CLA /DIV<DIVISOR?
1220  5247        JMP DV4 /YES
1221  7300        CLA CLL
1222  1256        TAD DCM15
1223  3257        DCA DIVCT
1224  5235        JMP DV2
1225  1253  DV3,  TAD HDIV
1226  7004        RAL
1227  3253        DCA HDIV
1230  1253        TAD HDIV
1231  1255        TAD DIV
1232  7430        SZL
1233  3253        DCA HDIV
1234  7200        CLA
1235  1254  DV2,  TAD LDIV
1236  7004        RAL
1237  3254        DCA LDIV
1240  2257        ISZ DIVCT
1241  5225        JMP DV3
1242  1253        TAD HDIV
1243  3600        DCA I DIVIDE
1244  2200        ISZ DIVIDE
1245  1254        TAD LDIV
1246  5600        JMP I DIVIDE
1247  1254  DV4,  TAD LDIV          /QUOTIENT=0, REMAINDER
1250  3600        DCA I DIVIDE      /=DIVIDEND
1251  2200        ISZ DIVIDE
1252  5600        JMP I DIVIDE
1253  0000  HDIV,  0
1254  0000  LDIV,  0
1255  0000  DIV,   0
1256  7763  DCM15, -15
1257  0000  DIVCT, 0

      /ZEROES MX BUF

1260  0000  ZMX,   0
1261  1273        TAD ZCM400
1262  3274        DCA CT400
1263  7240        CLA CMA
1264  3016        DCA R16
1265  6221        CDF2
1266  3416  NX400, DCA I R16
1267  2274        ISZ CT400       /DONE?
1270  5266        JMP NX400       /NO
1271  6201        CDF0
1272  5660        JMP I ZMX
```

```
1273  7400  ZCM400,  -400
1274  0000  CT400,   0

1275  0000  DOIT,    0
1276  3307          DCA CYL10
1277  1275          TAD DOIT
1300  1316          TAD CM3
1301  3315          DCA DTMP10
1302  1715          TAD I DTMP10
1303  1027          TAD DSEC
1304  3310          DCA HED10
1305  4442          JMS I XDISK
1306  0200  MEM10,   200
1307  0000  CYL10,   0
1310  0000  HED10,   0
1311  3400          3400
1312  4714          JMS I XDAND
1313  5675          JMP I DOIT

1314  1600  XDAND,   DANDY
1315  0000  DTMP10,  0
1316  7775  CM3,     -3

            /PROCESS BUSINESS-PROFESSIONAL-RESIDENTIAL SEGMENTS

1317  7201  BP,      CLA IAC          /BUS-PROF
1320  3033  RP,      DCA BPRADD       /RES-PROF
1321  1032          TAD ECHO
1322  7640          SZA CLA
1323  5342          JMP HIT0         /1ST FIELD PRESENT
1324  1033          TAD BPRADD
1325  1047          TAD XC234
1326  3417          DCA I R17
1327  3417          DCA I R17        /IF NO FINDING NAME
1330  4754          JMS I XD016      /SEARCH ALL SEGMENTS AT END
1331  1051          TAD XC1400
1332  3027          DCA DSEC
1333  7201          CLA IAC
1334  3007          DCA DRIVER
1335  4435          JMS I XGET16
1336  0001          1
1337  7201          CLA IAC
1340  3030          DCA UNIT         /SET TO UNIT 1 & GET HITS
1341  5434          JMP I XHIT
1342  1033  HIT0,    TAD BPRADD
1343  1047          TAD XC234
1344  3346          DCA TMP0
1345  4441          JMS I XDIVID
1346  0000  TMP0,    0
1347  0024          24
1350  0000          0        /HED
1351  1025          TAD XCYL/CYL+CYL STRT
1352  4443          JMS I XDOIT      /AND SPLIT XM
1353  5434          JMP I XHIT
1354  2253  XD016,   D016

            PAGE
```

```
1400  0000  DCSET,  0
1401  7300        CLA CLL
1402  1600        TAD I DCSET
1403  2200        ISZ DCSET
1404  6501        DCBRH   /SET COMMAND REGISTER
1405  7300        CLA CLL
1406  1600        TAD I DCSET
1407  2200        ISZ DCSET
1410  7106        CLL RTL
1411  7006        RTL
1412  6500        DCBRL   /WITH OP TO EXECUTE
1413  7300        CLA CLL
1414  5600        JMP I DCSET

            /DISK ROUTINES
            /CALL:  JMS I XDISK
            /       MEMORY BANK     (0-5) & DRIVE (6-11)
            /       CYLINDER
            /       SECTOR (0-5) & HEAD     (6-11)
            /       COMMAND

1415  0000  DISK,   0
1416  7300        CLA CLL
1417  1343        TAD DPRM+3
1420  3337        DCA DFAIL
1421  1615        TAD I DISK      /GET MEMORY BANK & DRIVE
1422  0341        AND DPRM+1
1423  3336        DCA DFIELD
1424  1007        TAD DRIVER
1425  3251        DCA DRIVE
1426  2215        ISZ DISK
1427  1615        TAD I DISK
1430  3254        DCA CYL
1431  2215        ISZ DISK
1432  1615        TAD I DISK
1433  0342        AND DPRM+2
1434  3273        DCA HED
1435  1615        TAD I DISK
1436  7002        BSW
1437  0342        AND DPRM+2
1440  3322        DCA SEC
1441  2215        ISZ DISK
1442  1615        TAD I DISK
1443  3321        DCA COM
1444  2215        ISZ DISK

1445  7305  DRTN,   CLA CLL IAC RAL
1446  6517        DMCR    /MODE=LOAD
1447  4200        JMS DCSET       /LOAD DRIVE #
1450  6000        6000
1451  0000  DRIVE,  0
1452  4200        JMS DCSET       /LOAD CYLINDER #
1453  1000        1000
1454  0000  CYL,    0
1455  4200        JMS DCSET       /SEEK & RESET HEAD
1456  1400        1400
1457  0060        60
1460  7325        CLA CLL CML IAC RAL
1461  6517        DMCR    /MODE=READ
1462  7300        CLL CLA
1463  6503        DSSRH
1464  0340        AND DPRM
1465  7640        SZA CLA /DRIVE READY?
1466  5263        JMP .-3 /NO
```

```
1467  7305          CLA CLL IAC RAL
1470  6517          DMCR    /MODE=LOAD
1471  4200          JMS DCSET       /LOAD HEAD
1472  0000          0
1473  0000  HED,     0
1474  1020          TAD WC  /PROCESS HEADER IMAGE
1475  3347          DCA HEAD+2      /WORD COUNT
1476  1021          TAD CA
1477  3351          DCA HEAD+4      /CURRENT ADDRESS
1500  1254          TAD CYL
1501  3346          DCA HEAD+1      /CYLINDER
1502  1273          TAD HED
1503  7002          BSW
1504  7110          CLL RAR
1505  1322          TAD SEC
1506  3345          DCA HEAD        /ALT, PROTECT, HEAD & SECTOR
1507  1345          TAD HEAD
1510  1346          TAD HEAD+1
1511  1347          TAD HEAD+2
1512  7041          CIA
1513  3350          DCA HEAD+3      /CHECKSUM
1514  1344          TAD HEADER
1515  6514          DDMAR   /SET CA TO HEADER
1516  7300          CLA CLL
1517  6516          DWCR    /SET WC=0
1520  4200          JMS DCSET       /LOAD READ/WRITE
1521  0000  COM,     0
1522  0000  SEC,     0
1523  7330          CLA CLL CML RAR
1524  6505          DAIRH   /ENABLE WRITE (4000
1525  1336          TAD DFIELD      /SET BUSY & MEM FIELD
1526  6512          DCSR
1527  6511          DESR0
1530  5327          JMP .-1
1531  7700          SMA CLA /ANY ERRORS ON DONE?
1532  5615          JMP I DISK      /NO
1533  2337          ISZ DFAIL       /ERRORS
1534  5245          JMP DRTN        /TRY AGAIN
1535  7402          HLT             /RETRY FAILURE

1536  0000  DFIELD, 0
1537  0000  DFAIL,  0
1540  0200  DPRM,   200
1541  7700          7700
1542  0077          77
1543  7770          -10

1544  1545  HEADER, HEAD
1545  0000  HEAD,   0       /HEAD, SECTOR, ETC.
1546  0000          0       /CYLINDER
1547  0000          0       /WORD COUNT
1550  0000          0       /CHECKSUM
1551  0000          0       /CURRENT ADDRESS
                            /
              PAGE

              /AND SUBROUTINE
```